

RAE

1. **TIPO DE DOCUMENTO:** Proyecto de Grado
2. **TITULO:** Diseño e Implementación de un Plugin en Formato Audio Unit, que Aplique el Fenómeno de Efecto Doppler a una señal de Audio.
3. **AUTOR:** CARDONA ARISTIZABAL, Daniel Eduardo.
4. **LUGAR:** Bogotá, Colombia.
5. **FECHA:** Junio 20, 2011.
6. **PALABRAS CLAVE:** Plugin, DSP, AudioUnit, Doppler, Efecto, SonicBirth.
7. **DESCRIPCIÓN DEL TRABAJO:** El siguiente trabajo fue realizado como proyecto de grado, muestra el proceso de desarrollo de un Plugin en formato Audio Unit para emular el efecto acústico "Doppler" a una señal de audio. Además se desarrolló una herramienta de medición para correlacionar los datos obtenidos con cálculos objetivos.
8. **LINEA DE INVESTIGACIÓN:** El proyecto está suscrito a las líneas de investigación de la Universidad de San Buenaventura tecnologías actuales y sociedad, ya que se basa en la implementación de un software tipo *plug-in*, para emular un efecto acústico, y que se utilizados por ingenieros en banda sonora y producciones musicales.
9. **FUENTES CONSULTADAS:** ALEXANDER ELLIS. On the Musical Scales of Various Nations. 1885. ANDY DUNCAN, DAVE ROSSUM. Fundamentos para Pitch Shifting, Presentado en la 85 convención, 1988. ANTONI CARRION ISBERT. Principios Básicos del Sonido, Diseño acústico de espacios arquitectónicos, 1998 APPLE INC. Audio Unit Programming Guide, 2007. JOSEPH MACHLIS. An Introduction to Perceptive Listening, The Enjoyment of Music. 1963. LISA GOTHARD, JOE ROSEN. Encyclopedia of Physical Science, Doppler Efect. 2009. p. 155. ROBERT RESNICK, DAVID HALLADAY. Física, Ondas Sonoras. Efecto Doppler. 1974.

10. **CONTENIDOS:** Este trabajo trata principalmente sobre desarrollo de software, en este caso para el procesamiento digital de señales de audio, el desarrollo de software es el contenido principal tanto para el desarrollo del Plugin como para la herramienta de medición del mismo. Además de esto muestra el análisis realizado a la fórmula de efecto Doppler y la correlación de los datos obtenido al utilizar el Plugin con cálculos objetivos.

11. **METODOLOGIA:** Este documento muestra el proceso desarrollado para realizar el diseño e implementación de un plugin de audio, específicamente en formato Audio Unit, que trabajara en editores de audio como Logic, Ableton Live, Audacity entre otros. El plugin emulara el efecto Doppler basándose en las ecuaciones originales del físico australiano Christian Doppler. En primer lugar se analizaron dos programas para el desarrollo de Audio Unit, X Code Tools de Apple y Sonic Birth del Politécnico de Montreal. Se decidió implementar el Audio Unit en Sonic Birth ya que se consideró más apropiado para un ingeniero de sonido. Posteriormente se procedió a diseñar el Audio Unit, sus parámetros y un boceto de la interfaz de usuario. Finalmente se realizó una herramienta de medición para la aplicación relacionando los datos obtenidos con cálculos objetivos.

12. **CONCLUSIONES:** El *Audio Unit* emula el efecto Doppler, por medio de un movimiento oscilatorio de la señal, relacionando el nivel con la distancia, al irse acercando la fuente, o aumentando el nivel, el tono percibido ha aumentado y va disminuyendo a media que la fuente se acerca, al estar el nivel original de la señal, o en frente del escucha, el tono es el mismo al original, y al alejarse, o reducir el nivel, el tono sigue disminuyendo, desde el original. La adición de un movimiento sincronizado con el tempo del *host* le agrega un poco de musicalidad al efecto. Se midió el cambio de tono producido por el *plug-in*, correlacionando los resultados con los datos obtenidos con los cálculos a diferentes velocidades de desplazamiento de la fuente, estableciendo una relación entre las velocidades de la fuente y del oscilador.

DISEÑO E IMPLEMENTACIÓN DE UN PLUGIN EN FORMATO AUDIO UNIT,
QUE APLIQUE EL FENÓMENO DE EFECTO DOPPLER A UNA SEÑAL DE
AUDIO

DANIEL EDUARDO CARDONA ARISTIZÁBAL

UNIVERSIDAD SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SONIDO
BOGOTÁ D.C.
2011

DISEÑO E IMPLEMENTACIÓN DE UN PLUGIN EN FORMATO AUDIO UNIT,
QUE APLIQUE EL FENÓMENO DE EFECTO DOPPLER A UNA SEÑAL DE
AUDIO

DANIEL EDUARDO CARDONA ARISTIZÁBAL

Proyecto de grado para optar por el título de Ingeniero de Sonido

Asesor de Fondo
Raúl Rincón

Asesor de Forma
Manuel Joves

UNIVERSIDAD SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SONIDO
BOGOTÁ D.C.
2011

Nota de la aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Bogotá, Mayo 2011

Dedicado a mi Mamá y a mi Abuela, quienes apoyaron
mis estudios como Ingeniero de Sonido,
y a mí Papá de quien heredé el amor por la música.

CONTENIDO

	pág.
INTRODUCCIÓN	16
1 PLANTEAMIENTO DEL PROBLEMA	17
1.1 ANTECEDENTES (ESTADO DEL ARTE)	17
1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA	18
1.3 JUSTIFICACIÓN	19
1.4 OBJETIVOS DE LA INVESTIGACIÓN	19
1.4.1 Objetivos Generales	19
1.4.1 Objetivos Específicos	19
1.5 ALCANCES Y LIMITACIONES DEL PROYECTO	20
1.5.1 Alcances	20
1.5.2 Limitaciones	20
2. METODOLOGIA	21
2.1 ENFOQUE DE LA INVESTIGACIÓN	21
2.2 HIPÓTESIS	21
2.3 VARIABLES	21
2.3.1 Variables Independientes	21
2.3.2 Variables Dependiente	21

3. LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA / NUCLEOS PROBLÉMICOS	22
4 MARCO DE REFERENCIA	23
4.1 MARCO TEORICO - CONCEPTUAL	23
4.1.1 Tono (<i>Pitch</i>)	23
4.1.1.2 Percepción del Pitch	23
4.1.1.3 Tono definido e indefinido	25
4.1.1.4 Etiquetando los Tonos	25
4.1.2 Efecto Doppler	27
4.1.2.1 Efecto Doppler cuando el receptor se encuentra en movimiento.	27
4.1.2.2 Efecto Doppler cuando la fuente se encuentra en movimiento.	29
4.1.2.3 Efecto Doppler fuente y observador en movimiento.	30
4.1.2.4 Efecto Doppler fuente sonora a gran velocidad.	31
4.1.2.5 Análisis de la fórmula para Efecto Doppler	32
4.1.3 Plug-in	34
4.1.4 Audio Units (AU).	35
4.1.4.1 Estructura programática y Ciclo de vida del A.U.	35
4.1.4.2 La <i>Audio Unit Specification</i> .	38
4.1.5 Funcionamiento del Audio Unit	39
4.1.5.1 Abrir y cerrar el Audio Units.	39
4.1.5.2 Inicialización múltiple.	39

4.1.5.3 Gráficos de procesamiento de Audio y el Modelo Pull (Arrastre).	39
4.1.6 Procesamiento.	42
4.1.6.1 Reiniciar - Resetting.	42
4.1.6.2 Ajustes en el Render (Impresión).	42
4.1.6.3 Automatización de Parámetros.	43
4.2 MARCO LEGAL	45
5. DESARROLLO INGENIERIL	46
5.1 DEFINICIÓN DE LA PLATAFORMA DE PROGRAMACIÓN SOBRE LA CUAL SE DESARROLLARA EL PLUG-IN	46
5.1.1 Análisis del desarrollo de Tremolo Unit en XcodeTools	46
5.1.2 Desarrollo de Reverberador de Schroeder en SonicBirth	47
5.1.3 Conclusiones Comparación	48
5.2 DISEÑO DEL ALGORITMO PARA EFECTO DOPPLER	49
5.2.1 Documento de Diseño	49
5.2.2 Análisis de la fórmula para el Doppler Pitch Shift	49
5.2.2.1 Análisis del cambio en la frecuencia de Doppler en relación a la velocidad del emisor, temperatura y frecuencia	49
5.2.2.2 Cambio en cents de la frecuencia Doppler	53
5.3 DISEÑO DE LOS PARÁMETROS DE LA INTERFAZ DE USUARIO (GUI)	55
5.3.1 Definición de los parámetros y sus presets.	55

DEFINICIÓN DEL TIPO DEL AUDIO UNIT	57
5.5 INFORMACIÓN DEL <i>BUNDLE</i> DEL AUDIO UNIT.	57
5.6 CREACIÓN DEL "CÓDIGO DE DESARROLLADOR".	58
5.7 BOCETO DEL AUDIO UNIT	58
5.8 DESARROLLO DEL ALGORITMO PARA EFECTO DOPPLER, EN SONICBIRTH.	59
5.8.1 Desarrollo del Audio Unit de Paneo y Ganancia.	59
5.8.1.1 Pad MIDI XY.	59
5.8.1.2 Osciladores de Nivel.	60
5.8.1.3 Paner y Oscilador de Paneo.	61
5.8.2 Pitch Shift.	63
5.8.2.1 Pitch Shift 1.	63
5.8.2.2 Pitch Shift 2.	64
5.8.3 Relación entre Velocidad, Nivel y Pitch de la señal para el efecto Doppler.	64
5.8.3.1 Calibración del Cambio Doppler para la velocidad de los osciladores.	64
5.8.3.2 Sincronización de osciladores de Pitch y Trayectoria.	67
5.8.3.3 Movimiento tipo "One Shot"	68
5.8.3.4 Opciones Doppler Start / Stop, Trigger / Reset y Sync Oscilators.	68
5.9 DESARROLLO DE LA INTERFAZ GRÁFICA DE USUARIO	70
6 PRESENTACIÓN Y ANÁLISIS DE RESULTADOS	72

6.1 ANÁLISIS DEL FUNCIONAMIENTO DEL DOPPLER EFFECT, EN AU LAB Y SONICBIRTH	72
6.1.1 Velocidad de procesamiento del Audio Unit en SonicBirth	73
6.1.2 Funcionamiento y control del Audio Unit en AU Lab.	73
6.1.3 Control de parámetros por MIDI en AU Lab y SonicBirth	74
6.2 ANÁLISIS DEL FUNCIONAMIENTO DEL AUDIO UNIT EN GARAGE BAND Y AUDACITY.	76
6.2.1 Automatización de Parámetros y Control MIDI	77
6.3 DESARROLLO DEL INSTRUMENTO MEDICIÓN PARA LA APLICACIÓN DESARROLLADA, CORRELACIONANDO LOS DATOS CON MEDICIONES OBJETIVAS.	77
6.3.1 Desarrollo de una herramienta de medición para el plug-in DC Doppler AU.	77
6.3.2 Medición del cambio Doppler para movimiento sencillo en el eje Y. Utilizando una frecuencia fundamental de 440 Hz.	78
6.3.3 Medición del cambio Doppler para movimiento sencillo en el eje Y. Utilizando una frecuencia fundamental de 1000 hz.	80
6.3.4 Tabla de cambio Doppler para el Oscilador de Nivel	81
6.3.5 Medición del cambio Doppler para movimiento sencillo en el eje X. Con una frecuencia fundamental de 1000 Hz.	81
6.3.6 Tabla de cambio Doppler producido por el <i>plug-in</i> con el oscilador de Paneo.	82
6.3.7 Medición del cambio Doppler obtenido con el <i>plug-in</i> , para una señal compleja, con movimiento sencillo en el eje Y.	83
6.3.7 Medición del cambio Doppler obtenido con el <i>plug-in</i> , para una señal compleja, con movimiento sencillo en el eje Y.	84
6.4 Correlación de los intervalos Doppler a diferentes velocidades obtenidos con el <i>plug-in</i> , con los obtenidos utilizando la ecuación.	85

7. CONCLUSIONES Y RECOMENDACIONES	87
7.1 CONCLUSIONES	87
7.2 RECOMENDACIONES	88
8. ERRORES SISTEMATICOS	89
9 BIBLIOGRAFÍA	90
GLOSARIO	91
ANEXOS	93

LISTA DE TABLAS

	pág.
Tabla 1. Frecuencia Doppler a 3 km/h. La fuente se acerca. Velocidad de la fuente positiva	50
Tabla 2. Frecuencia Doppler a 3 km/h. La fuente se aleja. Velocidad de la fuente negativa	50
Tabla 3. Frecuencia de Doppler a 30km/h.	51
Tabla 4. Frecuencia Doppler A 60 Km/h	52
Tabla 5. Frecuencia Doppler a 120 km/h.	52
Tabla 6. Frecuencia Doppler A 180 km/h.	53
Tabla 7. Cambio Doppler en Cents, a diferentes frecuencias	54
Tabla 8. Cambio Doppler en Cents, a diferentes temperaturas.	54
Tabla 9 Descripción de los parámetros del Audio Unit.	55
Tabla 10. Intervalo del cambio Doppler en Cents obtenidos al utilizar el plug-in, en un movimiento sencillo en el eje Y, a diferentes velocidades y frecuencias.	81
Tabla 11. Intervalo Doppler en el <i>plug-in</i> a diferentes velocidades del oscilador 2.	82
Tabla 12. Relación en entre las velocidades del Oscilador 1, con velocidades Km/h, por medio del el cambio en cents.	86
Tabla 13. Relación en entre las velocidades del Oscilador 2, con velocidades Km/h, por medio del el cambio en cents.	86

LISTA DE FIGURAS

	pág.
Figura 1. Frecuencias de notas, C en la cuarta octava, escala mayor diatónica, comenzando con C 1 (Do).	26
Figura 2. El efecto Doppler debido al movimiento del observador (oreja). La fuente se encuentra en reposo.	27
Figura 3. El efecto Doppler debido al movimiento de la fuente.	30
Figura 4. Efecto Doppler, fuente a muy alta velocidad.	31
Figura 5. Representación de un Audio Unit construido en SDK.	36
Figura 6. El Modelo Pull en acción con dos unidades de efectos.	40
Figura 7. Tremolo Unit desarrollado en XCode Tools.	46
Figura 8. Schroeder Reverberator desarrollado en SonicBirth.	48
Figura 9 Boceto de Doppler Effect.	58
Figura 10. Pad MIDI XY	54
Figura 11. Circuito para velocidades sincronizadas	60
Figura 12. Oscilador de ganancia	61
Figura 13. Circuito del Paner.	62
Figura 14. Circuito "Pitch Shift "1	63
Figura 15. Circuito "Pitch Shift "2	64
Figura 16. Diagrama para cambio de temperatura	65
Figura 17 Ejemplo función de cambio Pitch	67
Figura 18. Ejemplo de osciladores de nivel y pitch en sincronía	68
Figura 19. Ejemplo osciladores de paneo y pitch en sincronía, afecto por la posición del eje y la amplitud de la oscilación.	68

Figura 20. Comportamiento de los osciladores de pitch y nivel al acercase la fuente sonora.	69
Figura 21. Comportamiento de los osciladores de pitch y nivel al alejarse la fuente sonora.	69
Figura 22. Interfaz Gráfica de Usuario	71
Figura 23. Test de velocidad del circuito en SonicBirth	73
Figura 24. Controles MIDI disponibles en el Audio Unit	74
Figura 25. Doppler Effect en Garage Band	75
Figura 26. Carga de la CPU en Audacity a diferentes frecuencias de muestreo.	76
Imagen 27. Espectrograma para tono puro 440 hz.	78
Imagen 28. Vista de usuario del <i>plug-in</i> para la medición de un movimiento en eje Y en 440 Hz y 1kHz.	79
Imagen 29. Espectrograma movimiento sencillo 440 Hz	79
Imagen 30. Espectrograma movimiento sencillo, 1000 Hertz	80
Imagen 31. Vista de usuario del <i>plug-in</i> para la medición de movimiento en el eje X, para 1000 Hz.	81
Imagen 32. Cambio Doppler en eje X.	82
Imagen 33. Vista de usuario del <i>plug-in</i> para la medición de movimiento en el eje Y. Señal compleja.	84
Imagen 34. Vista de usuario del <i>plug-in</i> para la medición de movimiento en el eje Y. Señal compleja.	84
Imagen 35. Vista de usuario del <i>plug-in</i> para la medición de movimiento mixto en los dos ejes.	84
Imagen 36. Cambio Doppler movimiento mixto en 2 ejes	85

LISTA DE ECUACIONES

	pág.
Ecuación 1. Ecuación de tono para determinada frecuencia.	26
Ecuación 2. Ecuación para frecuencia Doppler 1.	28
Ecuación 3. Ecuación. Ecuación para frecuencia Doppler 2.	28
Ecuación 4. Ecuación. Ecuación para frecuencia Doppler, observador en movimiento.	28
Ecuación 5. Ecuación para frecuencia Doppler, observador en movimiento 2.	28
Ecuación 6. Ecuación para frecuencia Doppler, fuente acercándose al observador.	29
Ecuación 7. Ecuación para frecuencia Doppler, fuente alejándose del observador.	29
Ecuación 8. Ecuación para frecuencia Doppler, fuente en movimiento.	29
Ecuación 9. Ecuación para frecuencia Doppler, fuente y observador en movimiento.	30
Ecuación 10. Resultado figura 4.	32
Ecuación 11. Ecuación para receptor estático.	33
Ecuación 12. Ecuación para efecto Doppler receptor en movimiento.	33
Ecuación 13. Ecuación para efecto Doppler, ecuación generalizada.	33
Ecuación 14. Ecuación para efecto Doppler, movimiento lento de la fuente.	33
Ecuación 15. Ecuación utilizada en el diseño del del <i>plug-in</i>	49
Ecuación 16. Formula intervalo en cents entres frecuencia a y b.	53
Ecuación 17. Intervalo de pitch total en el AU.	66

LISTA DE ANEXOS

	pág.
Anexo 1. Diagrama para Oscilador 1	93
Anexo 2. Diagrama para Oscilador 2	94
Anexo 3. Diagrama de bloques para el Audio Unit de Doppler Effect, en SonicBirth.	95
Anexo 4 Herramienta de medición en MatLab	96
Anexo 5 Manual de Usuario	97
Anexo 5 Circuito del Reverberador de Schoeder, desarrollado en "SonicBirth"	100

INTRODUCCIÓN

Los *plug-ins* para audio, son un tipo de software (programa de computador), que generalmente no funcionan por sí mismos, sino que necesitan ser anclados dentro de otro software llamado estación de trabajo de audio digital (DAW), complementando sus funciones.

Pueden ser utilizados como instrumentos, efectos, entre otras utilidades para audio. Este tipo de software viene en diferentes formatos, para utilizar en los diferentes DAWs que existen actualmente en el mercado, principalmente RTAS para Pro Tools, VST para Cubase y Nuendo, Audio Units para Logic y Garage Band, (estos últimos solo pueden ser instalados en computadores Apple), y algunas veces en *Stand Alone*, lo que quiere decir que también funcionan por sí mismos.

Existen varios tipos de efectos, algunos están relacionados con el tiempo, como el "Delay" y "Reverb" y son comúnmente utilizados en sueños y recuerdos en banda sonora y muy utilizados en música para crear ambientes sonoros y dar espacialidad a determinados instrumentos; otros funcionan como modulación, variando las fases de la señal, como el "Flanger" y el "Phaser", entre otros.

En esta investigación, se desarrollará un *plug-in* en formato *Audio Unit*, de tipo efecto (*Audio FX*), que emulará el conocido efecto acústico, "Efecto Doppler". Este efecto es un fenómeno acústico que se produce por el desplazamiento de una fuente en relación a un observador, de un lugar a otro, donde se ven involucrados la ubicación en el espacio de una fuente (izquierda y derecha en Audio), nivel de presión sonora y altura tonal de ésta, se produce acústicamente, pero puede ser emulado mediante *software*, basándose en la teoría planteada por el físico Christian Doppler.

Es común escucharlo cuando un vehículo que tiene una sirena se acerca, pasa y se aleja de un observador. EL "*pitch*" o altura tonal, aumenta a comparación de la frecuencia emitida, a medida que se acerca, es idéntica cuando pasa frente al receptor y disminuye a medida que la fuente se aleja.

EL producto final de esta investigación será un *plug-in* formato AU, que procese el audio deseado aplicándole el Efecto Doppler a partir de unos parámetros predefinidos, para que sea utilizado y explotado por ingenieros de mezcla, productores, ingenieros de sonido y banda sonora, dando una herramienta para la producción de música, que poco ha sido utilizada y que sin duda llegará a ser de gran utilidad.

PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES (ESTADO DEL ARTE)

Existen, y se están desarrollando varios proyectos relacionados con *plug-ins*, por parte de estudiantes en la Universidad de San Buenaventura, pero ninguno con el formato *audio unit*.

El antecedente más cercano a este proyecto es el "Diseño y construcción de un paquete de *plug-ins* en formato *VST*", que está siendo realizado por Andrés Vázquez y David Delgado; también se está desarrollando el "Diseño de un algoritmo matemático de tipo *VST*, que convolucione una señal con la respuesta al impulso en recintos cerrados", de Bogotá por María M. Campo, Jorge E. Otero y Alexander Campos y finalmente está el "Diseño e implementación de un instrumento virtual Diseño de un instrumento (*VSTi*)", utilizando bancos de sonidos de un instrumento de cuerda folclórico Colombia (El Tiple), Miguel A. Olivares.

Existen varias versiones de este *plug-in* de efecto Doppler desarrolladas por compañías extranjeras, como el "ADopplerEn2"¹, diseñado por la compañía "AZ Audio", que actúa mediante paneos y retrasos modulados por un oscilador de baja frecuencia, solo funciona en Microsoft Windows, también la compañía Inglesa "Oli Larkin Headlines" tiene en el mercado un *VST plug-in* llamado "Space Station"², de Efecto Doppler, que tiene una interfaz de usuario mucho más interactiva y es programado utilizando los parámetros la ubicación inicial, ubicación final y velocidad de la fuente y es activado mediante un *trigger* o disparador MIDI.

Los dos más populares, y que se encuentran en varios formatos para Windows y Mac son los producidos por la compañía americana Waves,³ también funciona mediante un *trigger* para activar el procesamiento de la señal y tiene un costo bastante elevado; Finalmente, la compañía GRM Tools tiene un *plug-in* llamado "Doppler"⁴. Estos únicos que funcionan en el sistema operativo de Macintosh en formato AU. Todos estos *plug-ins* trabajan de maneras muy diferentes y poseen interfaces gráficas muy variadas.

El formato de *plug-in VST*, o *Virtual Studio Technology*, fue desarrollado por Steingber y es el más utilizado ya que posee una interfaz estándar. Funciona en varios software anfitriones o *VST Host* como Cubase, Frooty Loops Studio y Ableton Live. Pueden ser utilizados para procesar una señal o para generar señales en este caso son llamados *VST Instruments* o *VSTi*. Para desarrollar este tipo de *plug-ins* se puede utilizar el SDK de Steinger que se consigue en la

1 www.hitsquad.com/smm/programs/ADopplerEn/

2 www.olilarkin.co.uk/index.php?p=spacestation

3 www.waves.com/Content.aspx?id=244

4 www.grmtools.org/qt/files/Doppler.html

página web, aunque exciten programas de código abierto o *open source* para el desarrollo de este tipo de *plug-ins*.

El RTAS, de las siglas *Real Time Audio Suite* es un formato de *plug-in* desarrollado por Digidesign, actualmente Avid Technology, que funciona únicamente en las diferentes versiones de ProTools, (LE, HD, M-Powered). Pueden ser utilizados en tiempo real, como su nombre lo dice, o por medio de un render aplicado directamente al archivo de audio en ProTools. Para el desarrollo de *plug-ins* en este formato es necesario contactar a la compañía Avid por el SDK (kit de desarrollo de *software*). Hay restricciones en la publicación de información sobre los materiales de desarrollo de Avid, lo que hace el desarrollo de RTAS oficiales incompatible con *software* de código abierto (*open source*).

EL formato de *plug-in* "Audio Unit", fue desarrollado por la compañía Apple para sus programas editores de audio, como Garage Band, Logic entre otros. Por esta razón son los únicos que pueden ser utilizados únicamente en computadores de esta compañía dentro el sistema operativo Mac OS. Para el desarrollo de este *plug-in* se puede descargar el SDK directamente de la página de Apple, aunque existen programas de código abierto que permiten el desarrollo de *plug-ins* en este formato como SonicBirth.

También existe el formato DX de instrumentos y efectos que es un formato nativo para Windows PC, Pro-Tools TDM, el formato original para *plug-ins* basados en *hardware* de Pro-Tools, y los MOTU MAS, los *plug-ins* de *Mark Of The Unicorn*, y son un protocolo de transmisión de Digital Performer.

El Efecto Doppler es utilizado en diferentes áreas del audio, en producciones musicales, de géneros electrónicos y experimentales principalmente, en banda sonora para películas y sonido para video juegos. En Colombia, no existe ninguna compañía que diseñe este tipo de *software* o *plug-ins*, y a nivel mundial, son pocas las compañías que lo han desarrollado.

El *plug-in* diseñado en este proyecto, permitirá el usuario manipular los parámetros del efecto Doppler por medio de osciladores como el ADopplerEn2 de AZ Audio, y algunas de las versiones existente, en el desarrollo de este *software* el cambio Doppler de tono producido por el *plug-in*, será correlacionado con datos obtenidos con la fórmula de efecto Doppler. A diferencia de la mayoría de *plug-ins* que emulan efecto Doppler este estará en formato *audio unit*, y además será desarrollado dentro de un programa de código abierto.

1.2 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

Existen solo dos versiones de *software* que emulan el efecto Doppler en formato *audiounit*. El "Doppler" de la compañía GRM Topls y el "Doppler Effect" de Waves ambas son bastante buenas, y costosas aunque existen algunas versiones de *plug-in* para efecto Doppler en otros formatos bastante

incompletas, ya que no son claras en la manera en la que emulan el efecto y tiene interfaz de usuario poco "amigables", posiblemente no estén diseñadas en base a datos reales de efecto Doppler.

Cuando un ingeniero de audio pretende utilizar este "efecto", ya sea para mezcla o banda sonora, no cuenta con un software que emule de manera precisa el fenómeno deseado y mucho menos si trabaja en DAWs (*Digital Audio Workstation*) como Logic o Sound Track Pro.

¿Cómo desarrollar un *plug-in* en formato Audio Unit, que aplique el Efecto Doppler a una señal de audio?

1.3 JUSTIFICACIÓN

El propósito de este proyecto es innovar en materia de *software* para audio, trabajando con un efecto acústico que es de gran utilidad con, aplicaciones su en banda sonora, sonido para video juegos y producción musical.

Con este proyecto se pretende emular un efecto acústico producido por el movimiento de una fuente sonora, por medio de un *plug-in* que servirá para trabajar en diferentes estaciones de trabajo de audio digital y permitirá, a partir de la teoría existente acerca del comportamiento del efecto Doppler, modular y automatizar los parámetros básicos que se modifican en el movimiento de una fuente, como la ubicación de la fuente, la velocidad de desplazamiento y la altura tonal de la señal de audio.

Este proyecto se realizara porque actualmente existen muy pocas compañías que programen este tipo de *plug-ins* y, más aún, en formato AU. Es un *plug-in* innovador y es de gran ayuda para estudiantes, docentes, productores, ingenieros de mezcla y productores de banda sonora a la hora de buscar nuevas posibilidades para utilizar en su trabajo, en especial si pretenden simular de manera realista el movimiento de una fuente.

1.4 OBJETIVOS DE LA INVESTIGACIÓN

1.4.1 Objetivo General

- Desarrollar un *plug-in* tipo *Audio Unit*, para procesar una señal de audio, emulando el fenómeno de efecto Doppler.

1.4.2 Objetivos Específicos

- Determinar la plataforma de programación sobre la cual se desarrollara el *plug-in*.

- Diseñar el algoritmo para efecto Doppler.
- Desarrollar la interfaz de usuario.
- Realizar un instrumento de medición para la aplicación desarrollada, correlacionando los datos con mediciones objetivas.

1.5 ALCANCES Y LIMITACIONES DEL PROYECTO

1.5.1 Alcances

El impacto de este proyecto es el de innovar en materia de *software* para procesamiento de señales de audio, con un *plug-in* que tiene pocas versiones existentes a nivel mundial, algunas un poco incompletas ya que no son claras en la manera en que emulan el efecto, algunas tienen interfaces de usuario muy poco claras y no están correlacionadas con datos reales. Tan solo dos en formato *audio unit*, un formato en el cual es gratuito realizar la programación, ya que existe un gran apoyo por parte de la compañía Apple que utiliza *plug-ins* en este formato.

Con este *plug-in* se pretende brindar una herramienta a nivel de ingeniería de audio, utilizando un efecto acústico muy interesante que brindará una gran sensación de espacialidad y de dinámica en una canción, y que dará realismo en el desplazamiento de una fuente sonora en una producción musical o audiovisual. Será de gran utilidad para su aplicación de producciones musicales y banda sonora, principalmente.

1.5.2 Limitaciones

- Falta de bases previas por parte del desarrollador en el lenguaje de programación utilizado por la plataforma de desarrollo.
- EL tamaño y duración total del archivo de audio a procesar puede afectar considerablemente el tiempo de respuesta del *plug-in*, producir adición de latencia o sobrecarga de la CPU.
- El *software* sólo podrá ser utilizado en OSX, por programas que soporten el formato AU.

2. METODOLOGÍA

2.1 ENFOQUE DE LA INVESTIGACIÓN

Empírico-analítico, debido a que se realizará el diseño de un *software* tipo *plug-in*, que emule el fenómeno de efecto Doppler en una señal de audio, utilizando una interfaz gráfica que permitirá un manejo sencillo al usuario.

2.2 HIPÓTESIS

Se diseñará e implementará un *plug-in* tipo en formato *audio unit*, que aplique a una señal el fenómeno acústico llamado efecto Doppler, con una interfaz de usuario que permita elegir los parámetros básicos que se ven involucrados en este fenómeno.

2.3 VARIABLES

2.3.1 VARIABLES INDEPENDIENTES

- Método de funcionamiento, (Post-procesado o tiempo real).
- Software de desarrollo utilizado.
- Precisión de la emulación.
- La velocidad de desplazamiento de la fuente. (Variable, Constante).

2.3.2 VARIABLES DEPENDIENTES

- Hardware utilizado para realizar la programación.
- El lenguaje de programación utilizado por el software de desarrollo.

3. LÍNEA DE INVESTIGACIÓN DE USB / SUB-LÍNEA DE FACULTAD / CAMPO TEMÁTICO DEL PROGRAMA/NUCLEOS PROBLÉMICOS

El proyecto está suscrito a las líneas de investigación de la Universidad de San Buenaventura: Tecnologías actuales y sociedad, ya que se basa en la implementación de un software tipo *plug-in*, para emular un efecto acústico, y que se utilizados por ingenieros en banda sonora y producciones musicales.

Sub-línea de la Facultad: Procesamiento de señales digitales y-o analógicas, ya que se diseñará un software que module los parámetros de paneo, ganancia y altura tonal de la señal.

El campo de investigación del programa es Diseño de Sistemas de Sonido, debido a que se diseñará y desarrollará un algoritmo que permitirá implementar un fenómeno acústico a una señal de audio, que es de gran utilidad en banda sonora y mezcla musical.

4. MARCO DE REFERENCIA

4.1 MARCO TEORICO - CONCEPTUAL

4.1.1 Tono (*Pitch*). El *pitch* representa la frecuencia fundamental percibida en un sonido, es uno de los cuatro atributos principales de un sonido junto con nivel de presión sonora, timbre, y ubicación de la fuente sonora.⁵

El *pitch* permite la construcción de melodías, las alturas tonales son comparadas como más altas o más bajas, y son cuantificadas como frecuencias (Ciclos por segundo o *Hertz*), correspondiendo muy cercanamente a la frecuencia de las repeticiones de las ondas sonoras. Sin embargo, el *pitch* no es una propiedad física objetiva, sino un atributo psico-acústico subjetivo del sonido.

4.1.1.2 Percepción del *Pitch*. El *pitch* es una sensación subjetiva, en la cual un escucha asigna a tonos percibidos posiciones relativas en una escala musical, basado principalmente en las frecuencias en vibración.

La mínima diferencia percibida, el umbral en que un cambio es percibido, depende en la frecuencia del tono y es alrededor de 4.3 cents, 1 cent = 1/100 de semitono, o alrededor de 0.36 Hz dentro de las frecuencias 1.000 Hz y 2000 Hz, pero en el rango de frecuencias entre 62 Hz y 125 Hz el cambio mínimo percibido se encuentra entre los 40 cents o alrededor de 2 Hz.

La diferencia mínima de altura tonal percibida es probada reproduciendo 2 tonos rápidamente, uno después del otro, sucesivamente, y preguntando a un escucha si encuentra alguna diferencia entre los dos tonos. Esta diferencia mínima se vuelve más pequeña si los dos tonos son reproducidos simultáneamente ya que el oyente puede discernir entre las frecuencias reproducidas. El número total de cambios de altura tonales percibidos en el rango de la audición humana es alrededor de 1400, la cantidad total de notas en la escala temperada es de 120.

La altura tonal está relacionada con la frecuencia, pero no son equivalentes. La frecuencia es la medida científica de la altura tonal o *pitch*. Mientras que la frecuencia es objetiva, el *pitch* es completamente subjetivo. El *pitch* de tonos complejos es ambiguo, lo que significa que dos o más "*pitches*" diferentes pueden ser percibidos dependiendo de la dirección de atención de un observador. Las ondas sonoras como tal no tienen un *pitch*, sus oscilaciones pueden ser medidas para obtener una frecuencia fundamental, pero corresponde al cerebro humano asignarlas a las cualidades interna del tono.

⁵ Machlis, Joseph. An Introduction to Perceptive Listening, The Enjoyment of Music. 1963.

Mientras la frecuencia fundamental real puede ser precisamente determinada a través de mediciones físicas, esta puede diferir del *pitch* percibido, debido los sobre tonos, también conocidos parciales superiores, armónicos o no. El sistema humano de percepción auditiva también suele tener problemas distinguiendo diferencias de frecuencia entre notas bajo ciertas circunstancias. De acuerdo con la terminología acústica de la ANSI " *pitch* " es el atributo auditivo del sonido de acuerdo al cual los sonidos pueden ser establecidos en una escala de bajos a altos.

La percepción relativa del *pitch* puede ser engañada, resultando en una "ilusión aural", hay muchas de estas, como la paradoja del tritono, pero la más notable la escala de *Shepard*, donde se hace una secuencia discreta o continua de tonos, formada especialmente para sonar como si la secuencia continuara ascendiendo y descendiendo para siempre.

El *pitch* depende en menor medida del nivel de presión sonora (sonoridad o volumen) del tono, especialmente en frecuencias menores a 1000 Hz y superiores a 2000 Hz. El *pitch* de tonos bajos, disminuye aún más a medida que el nivel de presión sonora se incrementa.

Por ejemplo, un tono de 200 Hz con un alto nivel de presión sonora parecerá un semitono más abajo en *pitch*, que si es apenas audible. El *pitch* de tonos agudos, se incrementa a medida de que su nivel aumenta. En este orden la percepción del *pitch* es como cualquier otro de los sentidos humanos el cual responde a la intensidad del estímulo como se establece en la ley de *Weber-Fechner*. Sonidos más complejos que una onda senoidal pura, también tienen un *pitch* o altura tonal, como el habla y las notas musicales.

Muchos de estos sonidos son aproximadamente periódicos, y el tono percibido está directamente relacionado al periodo. Un tono compuesto por 2 señales senoidales de 1000 Hz y 1200 Hz tendrá 3 tonos. Dos tonos espectrales en 1000 Hz y 2000 Hz, derivado de frecuencias físicas de los tonos puros, y un tercer "tono virtual" en 200 Hz, derivado de la frecuencia de repetición de la forma de onda.

Estos también tienen un espectro que es (más o menos) una pila de armónicos, y el tono percibido está directamente relacionado con el espaciamiento de estos armónicos. El armónico más bajo de la pila es llamado la frecuencia fundamental, y su frecuencia está fuertemente correlacionada con el *pitch*, a pesar de que un fuerte tono puede ser percibido aun cuando la fundamental no se encuentre.

Históricamente, el estudio de percepción de tono, y específicamente el caso en que la fundamental no se encuentra, ha sido un problema central en la psico-acústica, y ha sido muy útil en la realización de pruebas de representación, procesamiento y percepción del sonido en el sistema auditivo.

4.1.1.3 Tono definido e indefinido. No todos los instrumentos musicales producen notas con *pitch* definido, los instrumentos de percusión son frecuentemente distinguidos por si tienen o no un *pitch* en particular.

Un sonido o nota de tono definido es en el que es posible o relativamente fácil de determinar el *pitch*. Los sonidos con un *pitch* definido tienen espectro de frecuencia armónico, o cercano a armónico.

Un sonido generado en cualquier instrumento produce muchos tipos de vibraciones que ocurren simultáneamente. Un escucha oye muchas frecuencias de una sola vez. La vibración que tiene la más lenta frecuencia será llamada la fundamental. Las otras frecuencias son sobre tonos. Una case importante de sobre tonos es la formada por los armónicos, que tienen frecuencias que son múltiplos de la fundamental. Un sonido o nota de *pitch* indefinido es uno en que es imposible o relativamente difícil deducir su tono. Sonidos con tono indefinido no tiene un espectro armónico o tiene un espectro armónico alterado.

Aun así es posible para dos sonidos de tono indefinido ser claramente más altos o bajos en relación, por ejemplo, un redoblante tiene claramente un *pitch* más alto que un bombo, aunque los dos tienen un tono indefinido, porque su espectro tiene frecuencias más altas. En otras palabras es posible y frecuentemente fácil, definir los *itches* relativos de dos sonidos de *pitch* indefinido, pero cualquier sonido de *pitch* indefinido no corresponde perfectamente a un valor definido de *pitch*.

Un tipo especial de *pitch* frecuentemente ocurre en la naturaleza libre cuando el sonido de una fuente sonora alcanza el oído de un observador directamente y también después de haber sido reflejado en una superficie reflectora, Este fenómeno es llamado repetición de tono, porque se adiciona una repetición verdadera al sonido original, siendo este el pre-requisito básico.

4.1.1.4 Etiquetando los Tonos. El *pitch* puede ser etiquetado utilizando letras, como en la notación de *pitch de helmholtz*, o utilizando una combinación de letras y números, como en la notación científica de *pitch*, o por un número representando la frecuencia. Por ejemplo, podría referirse al "La" por encima del "Do" central, como A4 o A440. En la temperación estándar de occidente, el concepto de *pitch* es indiferente a la escritura. G4 con doble sostenido equivale a A4.

La percepción humana es aproximadamente logarítmica con respecto a la frecuencia fundamental, la distancia percibida entre los tonos A220 y A440 es la misma que la percibida entre A440 y A880.

Motivado por esta percepción logarítmica, la teoría musical algunas veces representa los tonos utilizando una escala numérica basada en el logaritmo de la fundamental. Por ejemplo, se puede adoptar el ampliamente utilizado

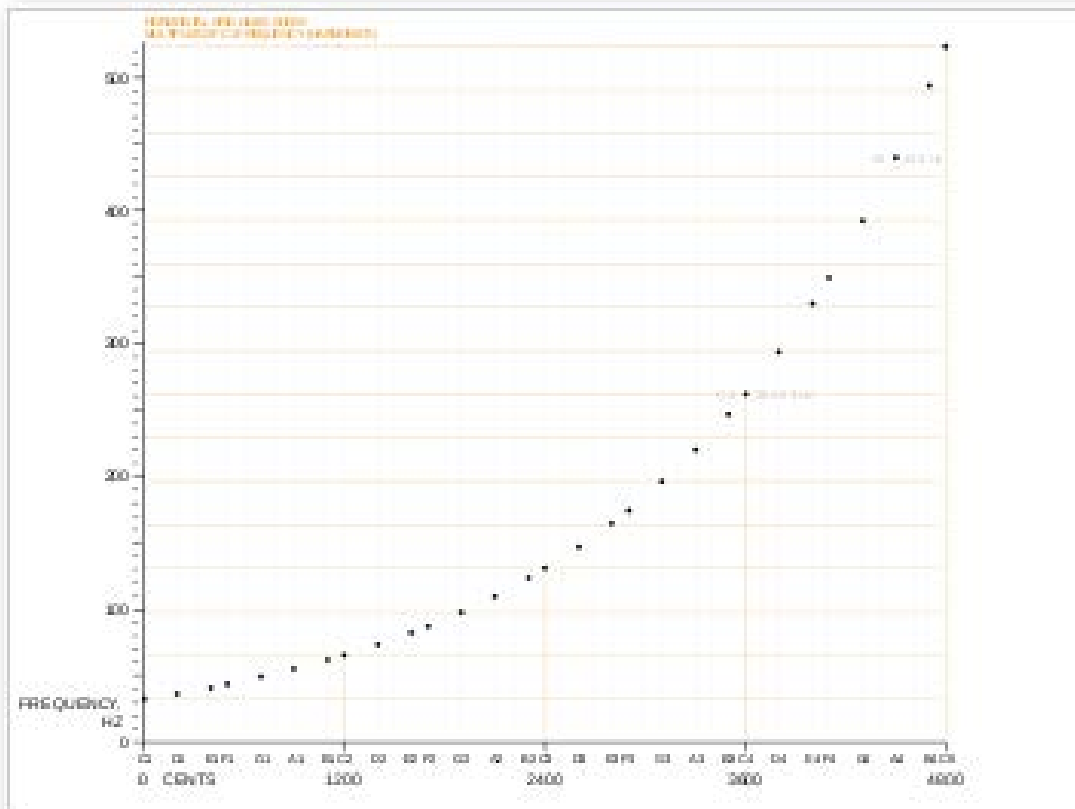
sistema MIDI para asignar la frecuencia fundamental f a un número real como se muestra.

Ecuación 1. Ecuación de tono para determinada frecuencia.

$$p = 69 + 12 \times \log_2 \left(\frac{f}{440 \text{ Hz}} \right)$$

Esto crea un espacio tonal lineal en el cual las octavas tienen 12 semitonos (la distancia entre las notas adyacentes del piano), el tono A 440 es asignado al número 69. La distancia en este espacio corresponde a la distancia musical medida en experimentos psicológicos, y comprendida por músicos. (Un semitono temperado es subdividido en 100 cents). El sistema es lo suficientemente flexible para incluir "micrótonos" que no se encuentran en un piano estándar. Por ejemplo el tono intermedio entre C (60) y C # (61) puede ser llamado 60.5.⁶

Figura 1. Frecuencias de notas, C en la cuarta octava, escala mayor diatónica, comenzando con C 1 (Do)



An Introduction to Perceptive Listening.

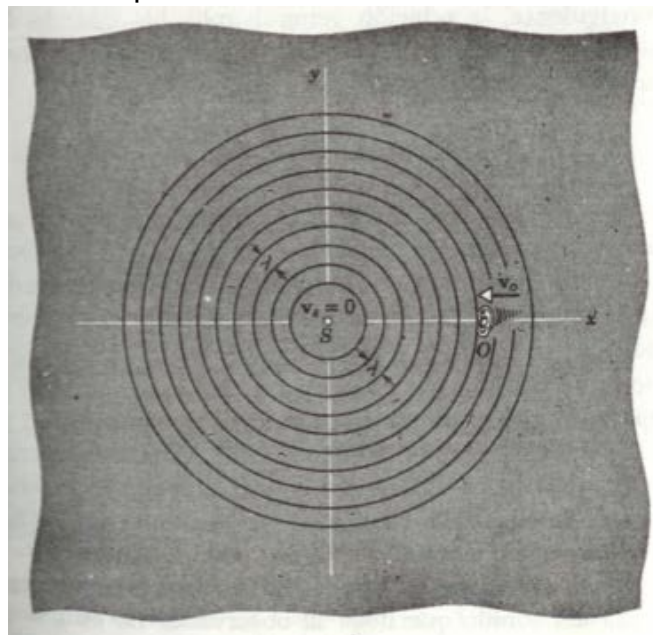
⁶ Machlis, Joseph. An Introduction to Perceptive Listening, The Enjoyment of Music. 1963.

4.1.2 Efecto Doppler. Cuando un observador que escucha se mueve hacia una fuente sonora en reposo, la altura (frecuencia) del sonido que se percibe es superior que cuando se encuentra en reposo. Si el observador se está alejando de la fuente fija, percibe un sonido más bajo que cuando se halla en reposo. Se obtienen resultados similares cuando la fuente se encuentra en movimiento, acercándose o alejándose de un observador en reposo.⁷

Christian Johann Doppler (1803-1853), de nacionalidad austríaca, en una memoria que presentó en 1842, llamó la atención al hecho de que el color de un cuerpo luminoso, lo mismo que la altura de un cuerpo sonoro, debe cambiar como consecuencia del movimiento relativo del cuerpo y del observador. Este efecto Doppler, que es el nombre que recibe, se aplica a toda clase de ondas en general. Se aplicara en este proyecto a ondas sonoras.

4.1.2.1 Efecto Doppler cuando el emisor se encuentra en movimiento. La figura 2 muestra una fuente sonora S en reposo en este marco y un observador O (nótese la oreja) que se mueve hacia la fuente con una velocidad v_o . Los círculos representan frentes de onda, espaciados a distancias de una longitud de onda, que van avanzando por el medio. Si el observador estuviera en reposo en el medio recibiría t/λ ondas en el tiempo t , siendo v la velocidad del sonido en el medio y λ la longitud de onda. Ahora bien, debido a su movimiento hacia la fuente, recibe $v_o * t/\lambda$ ondas *adicionales* en este mismo tiempo t .

Figura 2. El efecto Doppler debido al movimiento del observador (oreja). La fuente se encuentra en reposo



Física, Ondas Sonoras. Efecto Doppler

⁷ RESNICK, Robert; HALLADAY, David. Física, Ondas Sonoras. Efecto Doppler. 1974.

La frecuencia ν' que percibe es el número de ondas que recibe por unidad de tiempo.

Ecuación 2. Ecuación para frecuencia Doppler 1.

$$\nu' = \frac{vt/\lambda + v_o t/\lambda}{t} = \frac{v + v_o}{\lambda} = \frac{v + v_o}{v/\nu}$$

Es decir que:

Ecuación 3. Ecuación. Ecuación para frecuencia Doppler 2.

$$\nu' = \nu \frac{v + v_o}{v} = \nu \left(1 + \frac{v_o}{v} \right)$$

La frecuencia ν' que percibe el observador es igual a la frecuencia original ν que percibe en reposo más el aumento $\nu(v_o/v)$ que se produce por el movimiento del observador. Cuando éste se encuentra en movimiento alejándose de la fuente sonora, hay una disminución de frecuencia $\nu(v_o/v)$ correspondiente a las ondas que no llegan al observador en cada unidad de tiempo debido a su movimiento de alejamiento.

Ecuación 4. Ecuación. Ecuación para frecuencia Doppler, observador en movimiento.

$$\nu' = \nu \left(\frac{v - v_o}{v} \right) = \nu \left(1 - \frac{v_o}{v} \right)$$

Por consiguiente, la relación general aplicable cuando la fuente se encuentra en reposo con respecto al medio pero el observador se está moviendo a través del medio es:

Ecuación 5. Ecuación para frecuencia Doppler, observador en movimiento 2.

$$\nu' = \nu \left(\frac{v \pm v_o}{v} \right)$$

En la cual, el signo más es aplicable al movimiento hacia la fuente y el signo menos, al movimiento alejándose de la misma. La causa del cambio en este

caso es el hecho de que el observador intercepta más o menos ondas cada segundo debido a su movimiento por el medio.

4.1.2.2 Efecto Doppler cuando la fuente se encuentra en movimiento.

Cuando la fuente se encuentra en movimiento hacia un observador en reposo, el efecto es una reducción en la longitud de onda, porque la fuente está avanzando detrás de las ondas que se acercan al observador y, por consiguiente, las crestas quedan más próximas. Si la frecuencia de la fuente es ν y su velocidad v_s , durante cada vibración avanza una distancia v_s/ν y cada longitud de onda se reduce en esa cantidad. Por consiguiente, la longitud de onda del sonido que llega al observador no es $\lambda = v/\nu$ sino $\lambda' = v/\nu - v_s/\nu$. Por lo tanto, la frecuencia del sonido que percibe el observador aumenta.⁸

Ecuación 6. Ecuación para frecuencia Doppler, fuente acercándose al observador.

$$\lambda_1 = \frac{v_1}{\nu} = \frac{(\nu - \nu^s)\lambda}{\nu} = \lambda \left(\frac{\nu - \nu^s}{\nu} \right).$$

Si la fuente se mueve alejándose del observador, la longitud de onda emitida v_s/ν mayor que λ , de manera que el observador percibe una frecuencia reducida.

Ecuación 7. Ecuación para frecuencia Doppler, fuente alejándose del observador.

$$\lambda_1 = \frac{(\nu + \nu^s)\lambda}{\nu} = \lambda \left(\frac{\nu + \nu^s}{\nu} \right).$$

Por consiguiente, la relación general aplicable cuando el observador se encuentra en reposo con respecto al medio pero la fuente se está moviendo a través de él es. Expresión en la cual el signo menos es aplicable cuando el movimiento es hacia y el signo más cuando el movimiento es alejándose del observador.

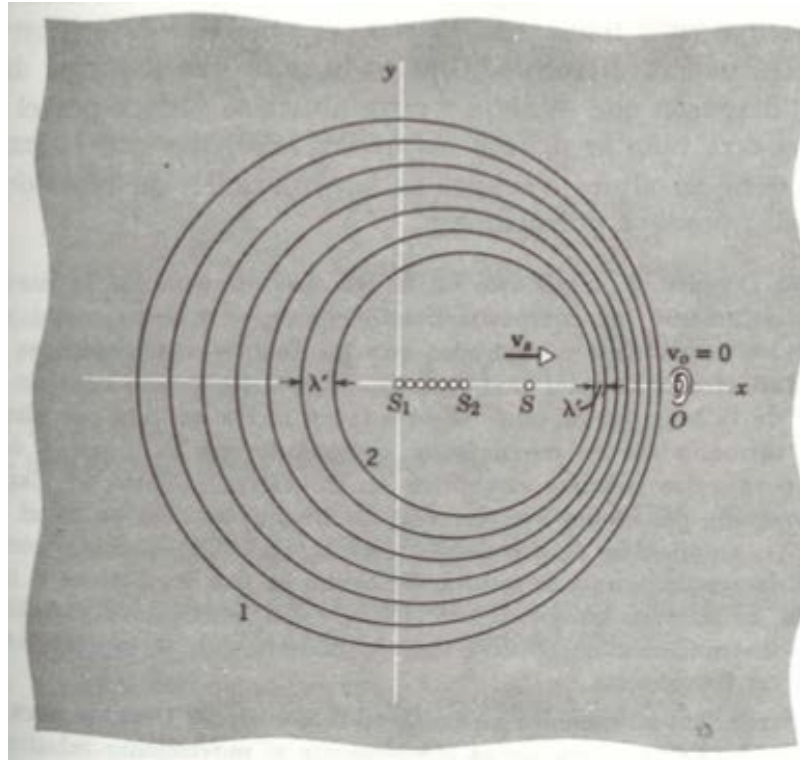
Ecuación 8. Ecuación para frecuencia Doppler, fuente en movimiento.

$$\lambda_1 = \lambda \left(\frac{\nu \pm \nu^s}{\nu} \right),$$

⁸ RESNICK, Robert; HALLADAY, David. Física, Ondas Sonoras. Efecto Doppler. 1974.

La causa del cambio en este caso es el hecho de que el movimiento de la fuente a través del medio reduce o aumenta la longitud de onda transmitida a través del medio.

Figura 3. El efecto Doppler debido al movimiento de la fuente.



Física. Ondas Sonoras. Efecto Doppler

El observa- se encuentra en reposo. El frente de onda 1 fue emitido por la fuente. Cuando ésta se encontraba en S_1 , el frente de onda 2 fue emitido cuando se encontraba en S_2 , etc. En el instante en que se tornó la "fotografía instantánea", la fuente se encontraba en S .

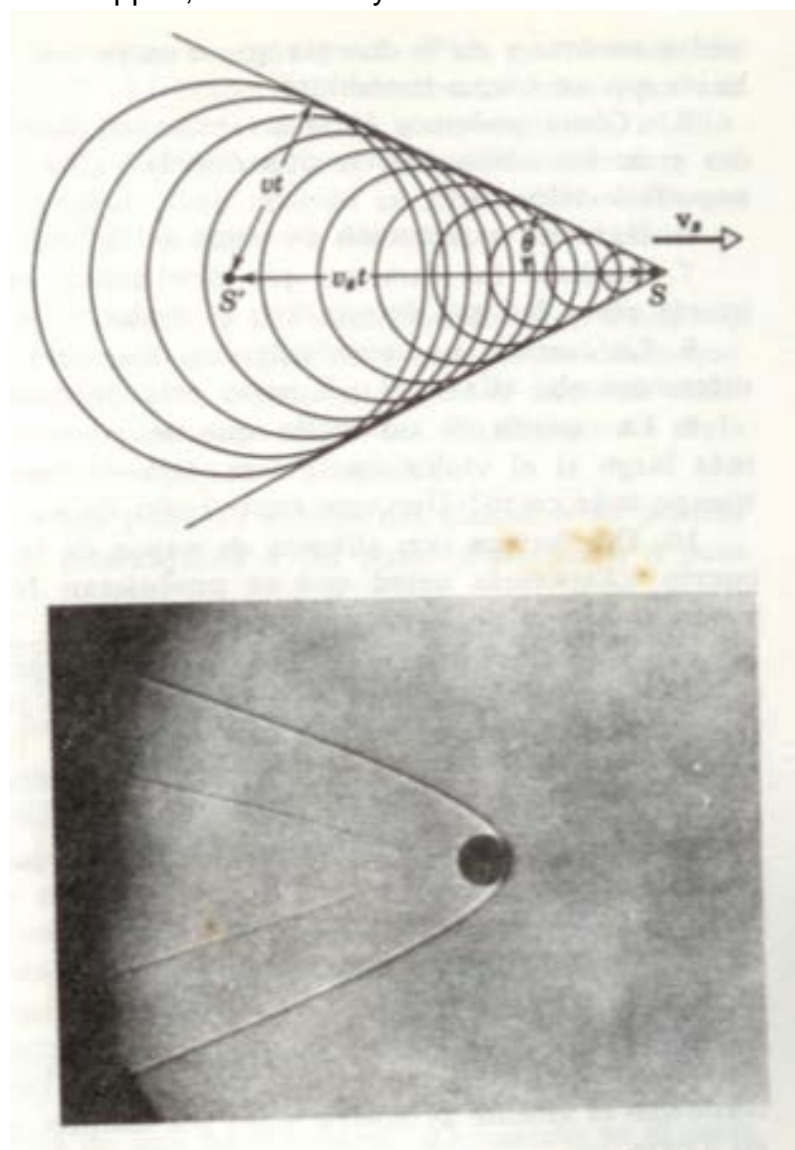
4.1.2.3 Efecto Doppler, fuente y observador en movimiento. En la siguiente ecuación, los signos superiores (+ en el numerador - en el denominador) corresponden al caso en que la fuente y el observador se mueven sobre la línea que los une, dirigiéndose uno hacia el otro, y los signos inferiores cuando se mueven alejándose uno del otro.

Ecuación 9. Ecuación para frecuencia Doppler, fuente y observador en movimiento.

$$v' = v \left(\frac{v \pm v_o}{v \mp v_s} \right)$$

4.1.2.4 Efecto Doppler, fuente sonora a gran velocidad. Hay muchos casos en los cuales la fuente se mueve a través de un medio con una velocidad mayor que la velocidad de fase de la onda en ese medio. En tales casos, el frente de onda toma la forma de un cono, en cuyo vértice está el cuerpo en movimiento. Algunos ejemplos son la onda de proa de un bote veloz en el agua y la "onda de choque" de un avión o proyectil que se mueve a través del aire con una velocidad mayor que la velocidad del sonido en ese medio (velocidades supersónicas). La radiación Cerenkov consiste en ondas luminosas emitidas por partículas cargadas que se mueven a través de un medio con una velocidad mayor que la velocidad de fase de la luz en ese medio.⁹

Figura 4. Efecto Doppler, fuente a muy alta velocidad.



⁹ RESNICK, Robert; HALLADAY, David. Física, Ondas Sonoras. Efecto Doppler. 1974.

Física, Ondas Sonoras. Efecto Doppler

En la imagen de arriba, un grupo de frentes de onda que acompañan a un proyectil que se mueve con velocidad supersónica. Los frentes de onda son esféricos y su envolvente es un cono. Abajo, una fotografía de chispa de un proyectil que tiene este movimiento. (Fotografía de la U. S. Navy).

En la figura 4 se representan las posiciones que en un momento dado tienen las ondas esféricas que fueron emitidas en diversas posiciones por la fuente durante su movimiento. El radio de cada esfera, en ese instante, es el producto de la velocidad de la onda v y del tiempo t que ha transcurrido desde que la fuente estaba en todo su centro. La envolvente de estas ondas es un cono cuya superficie forma un ángulo θ con relación a la dirección del movimiento de la fuente.

Ecuación 10. Resultado figura 4.

$$\text{sen}\theta = v/vs$$

Para las ondas en el agua, el cono se reduce un par de líneas que se intersectan. En aerodinámica, la relación vs/v se llama número de Match.

4.1.2.5 Análisis de la fórmula para Efecto Doppler. La frecuencia del sonido que la fuente emite no cambia realmente. Se considera la siguiente analogía, una persona arroja una pelota cada segundo en la dirección de otra persona. Asuma que la pelota viaja con velocidad constante. Si el que arroja se mantiene en un solo lugar, la otra persona recibirá una pelota cada segundo. Sin embargo, si el que arroja se acerca a la otra persona recibirá las pelotas más constantemente, porque estas estarán menos espaciadas entre sí.

El inverso es posible si el lanzador se aleja de la otra persona. Así que es la longitud de onda la que es afectada y por lo tanto la frecuencia percibida también es afectada. La velocidad de propagación de las ondas permanece constante, aunque la longitud cambio, por lo tanto la frecuencia también cambia.¹⁰

Si la fuente se aleja del observar emitiendo ondas a través de un medio con una frecuencia f_0 , entonces un observador estacionario relativo al medio detecta una frecuencia f dada por:

¹⁰ ROSEN, Joe; GOTHARD, Lisa Quinn. Encyclopedia of Physical Science. Doppler Effect. 2009. p. 155.

Ecuación 11. Ecuación para receptor estático.

$$f = \left(\frac{v}{v + v_s} \right) f_0$$

Donde v_s (Velocidad de la Fuente) es positiva si la fuente se aleja del observador y negativa si se acerca.

Un análisis similar para un observador en movimiento y una fuente estacionaria muestra la frecuencia observada, (la velocidad del receptor siendo representada como v_r) :

Ecuación 12. Ecuación para efecto Doppler receptor en movimiento.

$$f = \left(\frac{v + v_r}{v} \right) f_0$$

Donde una convención similar aplica: v_s es positiva si el observador se acerca a la fuente y negativa si este se aleja de la misma. Esto puede ser generalizado en una sola ecuación con ambas la fuente y el receptor en movimiento.

Ecuación 13. Ecuación para efecto Doppler, ecuación generalizada.

$$f = \left(\frac{v + v_r}{v + v_s} \right) f_0$$

Con una fuente que se mueve relativamente lenta, $v_{s,r}$ es pequeña en comparación a v y la ecuación se aproxima a:

Ecuación 14. Ecuación para efecto Doppler, movimiento lento de la fuente.

$$f = \left(1 - \frac{v_{s,r}}{v} \right) f_0$$

Donde

$$v_{s,r} = v_s - v_r.$$

Sin embargo las limitaciones mencionadas antes aun aplican. Donde una ecuación más complicadas es derribada si el uso de aproximaciones (solo asumiendo que la fuente, receptor y la onda se mueven linealmente entre sí)

muchos resultados interesantes y quizás sorprendentes son encontrados. Por ejemplo, como Lord Rayleigh anoto en su clásico libro en sonido, si se logra mover correctamente, una sinfonía puede ser escuchada en reversa. Esto también es llamado “Efecto de tiempo en reversa” del efecto Doppler.

Otra conclusión interesante es que el efecto Doppler es dependiente del tiempo (No solo necesite conocerse las velocidades de la fuente y el receptor, sino también sus posiciones en un tiempo dado), y además, en algunas circunstancias es posible recibir dos señales u ondas de una fuente, o ninguna señal. En adición hay más posibilidades que solo un receptor que se acerca a la señal y el receptor se alejen de la señal.

4.1.3 Plug-in. En cómputo, los *plug-ins* son un set de componentes de *software* que agregan unas habilidades específicas a una aplicación más grande. Si son soportados, los *plug-ins* permiten personalizar la funcionalidad de una aplicación. Por ejemplo, los *plug-ins* son comúnmente usados en navegadores de Internet para reproducir video, buscar virus o mostrar ciertos tipos de archivos.

Existen varios formatos de *plug-ins* para audio, este varía dependiendo del editor de audio en el que trabajen, o si están desarrollados para funcionar con algún tipo de hardware. Permiten personalizar los editores de audio agregando funciones como efectos y controlar instrumentos virtuales entre otra. Los principales formatos de *plug-ins* para audio son:

VST: La tecnología de estudio virtual de Steinberg, es una interface para integrar *samplers*, sintetizadores y efectos a algunos editores de audio que soportan este formato, como Cubase, Ableton Live, Audacity entre otros. Pueden ser clasificados como VSTi para los instrumentos y VST para efectos.

Audio: Units. Son una arquitectura proveída a nivel de sistema de complemento, que funciona con *Core Audio* en Mac OSX y desarrollada por computadores Apple. Se puede pensar que esta arquitectura de Apple es equivalente a otro formato popular para *plug-ins* de audio, el VST de Apple, ya que existen muchas similitudes entre los *Audio Units* y VST, existen muchas herramientas de conversión entre estos dos formatos. (*Symbiosis, FXpasin VST-AU Adapter*)

RTAS, TDM: De las siglas *Real Time Audio Suite* es un formato de *plug-in* desarrollado por Digidesign, actualmente *Avid Technology*, que funciona únicamente en las diferentes versiones de ProTools, (LE, HD, M-Powered). Pueden ser utilizados en tiempo real, como su nombre lo dice, o por medio de un render aplicado directamente al archivo de audio en ProTools. Para el desarrollo de *plug-ins* en este formato es necesario contactar a la compañía Avid por el SDK (kit de desarrollo de *software*). Hay restricciones en la publicación de información sobre los materiales de desarrollo de Avid, lo que hace el desarrollo de RTAS oficiales incompatible con software de código

abierto (*open source*). Pro-Tools TDM, es el formato original para *plug-ins* basados en *hardware* de Pro-Tools.

MAS: Los *plug-ins* de *Mark Of The Unicorn*, son un protocolo de transmisión de Digital Performer.

4.1.4 Audio Units (AU). Un *Audio Unit* (Unidad de Audio), comúnmente abreviado como AU, es un “*plug-in*” para *Mac OSX* que complementa aplicaciones de audio digital tales como *Logic* y *Garage Band*, también se puede utilizar los *Audio Units* para agregar características exclusivas a sus aplicaciones. En lenguaje de programación, un *Audio Unit* es empaquetado como un *Bundle*, (paquete o manojó de archivos) y configurado como un componente.¹¹

En un nivel más profundo, y dependiendo de su punto de vista, un *Audio Unit* puede ser dos cosas muy diferentes. Desde adentro, como es visto por un desarrollador de *Audio Units*, un AU es la implementación de un código ejecutable dentro de una *Plug-in Application Programming Interface*. LA *API* es estándar para que cualquier aplicación diseñada para trabajar con *Audio Units* sepa cómo utilizarla.

Un desarrollador de *Audio Units* puede agregar la posibilidad a usuarios o aplicaciones de controlar un *Audio Unit* en tiempo real a través del mecanismo de los parámetros del *Audio Unit*. Los parámetros son auto-descriptores, sus valores y capacidades son visibles para las aplicaciones que usen estos *Audio Units*.

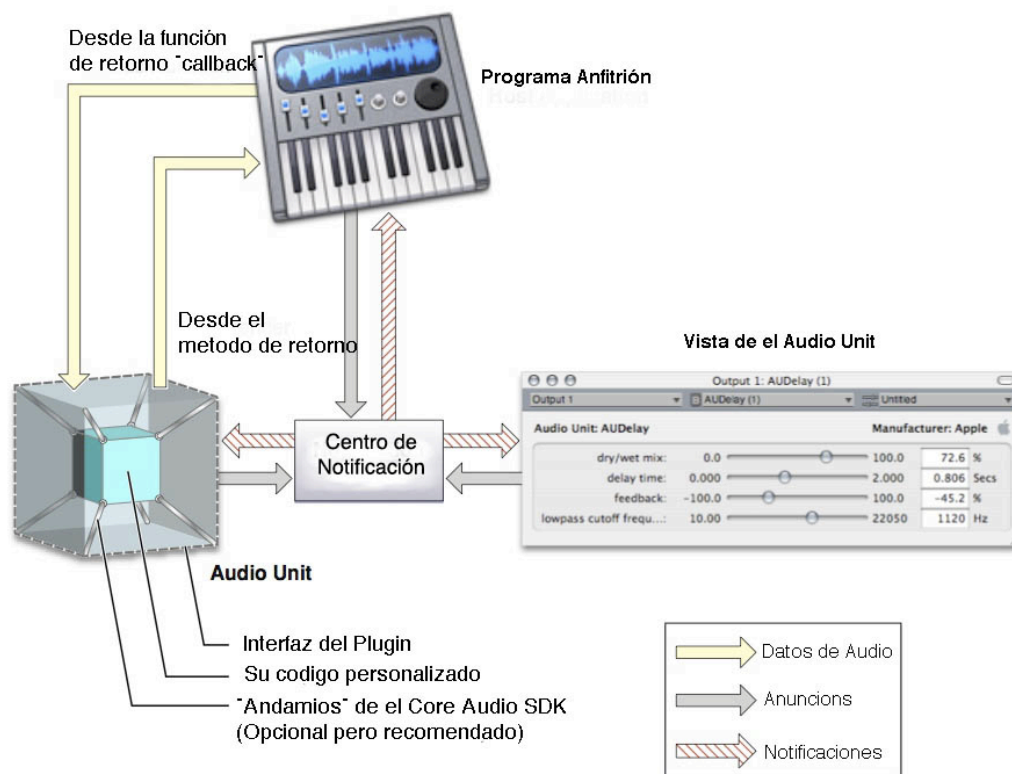
Desde afuera y como es visto por una aplicación que usa el *Audio Unit*, el AU es solo su *Plug-in API*. Esta *Plug-in API* permite a la aplicación “interrogar” el *Audio Unit* acerca de sus características específicas, definidas por el programador como parámetros y propiedades.

Debido a esta “encapsulación”, la manera en que se desarrolle el *Audio Unit* depende del programador. La manera más rápida, la que es presentada por Apple en la *Audio Unit Programming Guide* y descrita en este documento, es “sub clasificar”, las correctas “superclases C++” del *Core Audio SDK*, que se puede descargar gratuitamente en la página de Apple.

4.1.4.1 Estructura programática y Ciclo de vida del AU. La figura 5 representa un *Audio Unit* construido en SDK. La figura muestra el *Audio Unit* en contexto con su interfaz de usuario y con otra aplicación, conocida como *Host* (Anfitrión), que estará utilizando el AU.

¹¹ APPLE INC. Audio Unit Development Fundamentals, Audio Unit Programming Guide. 2007.

Figura 5. Representación de un Audio Unit construido en SDK



Audio Unit Programming Guide, Apple.

La figura muestra dos distintas partes internas del *Bundle* del *Audio Unit*, el AU como tal, a la izquierda, y la vista del Audio Unit a la derecha. El Audio Unit realiza el proceso al audio. La vista provee una interfaz gráfica de usuario para el AU, y si el programador lo provee, un soporte para automatización. Cuando se crea un AU, normalmente se empaquetan las dos partes en un mismo "Bundle" o manojos de archivos, como se mostrara más adelante, pero son piezas de códigos separadas lógicamente.

El AU, su interfaz y el Host se están comunicando entre ellos a través de un centro de notificación, establecido por el Host. Esto permite a las tres entidades permanecer sincronizadas. Las funciones para el centro de notificación son parte de la *Core Audio Audio Unite Event API*.

Cuando un usuario ejecuta por primera vez una aplicación anfitriona, ni el AU, ni su interfaz es utilizada. En este punto ninguna de las partes mostradas en la figura 5 están presentes a excepción de la aplicación Host.

El AU y su interfaz entran en acción con alguna de las dos siguientes maneras:

Típicamente, un usuario indica al Host que quiere utilizar un Audio Unit. Por ejemplo, un usuario podría pedir a la aplicación que agregue a un canal de audio un efecto de reverberación.

Para un AU que se desarrolla para agregar una característica a una aplicación, la aplicación abre el AU directamente, y probablemente en el momento en que se ejecuta la aplicación.

Cuando el *Host* abre el *Audio Unit*, lo engancha a la cadena de datos del *Host*, representada en la figura por la flecha amarilla clara (Datos de Audio). Este enganchamiento tiene dos partes, proveer datos de audio frescos a el *Audio Unit*, y retornar los datos de audio procesados desde el *Audio Unit*.¹²

- Para proveer los datos de audio nuevos a un AU, el *Host* define una función de *Callback* (ser llamado por el AU), que provee los datos de audio, de a una porción a la vez. Una **porción** es un número de cuadros de datos de audio. Un **cuadro** es una muestra de datos de audio a través de todos los canales.
- Para retribuir datos de audio procesados de un AU, El *Host* invoca el método de *render* del AU.

A continuación se muestra como sucede el flujo de datos de audio procesados entre la aplicación huésped y el AU.

1. El huésped invoca el método de *render* del AU, pidiendo al *Audio Unit* una muestra de audio procesado.
2. EL A.U. responde llamando a la función de *Callback* para obtener una muestra de datos de audio para procesar.
3. El *Audio Unit* procesa las muestras de datos de audio y muestra el resultado en un buffer de salida, para que los reciba el anfitrión.
4. El anfitrión recupera los datos y de nuevo invoca el método de *render* del AU.

En la representación del *Audio Unit* en la figura 8 el cubo exterior representa la *Plugin API*. Apple provee la *Audio Unit Specification*, que define la *Plugin API* para una variedad de tipos de *Audio Units*. Cuando se desarrolla el AU siguiendo esta especificación, este podrá trabajar con cualquier aplicación que siga la especificación.

Adentro, un AU contiene andamios programáticos que conectan el Plug-in API al código personalizado. Cuando se utiliza el *Core Audio SDK* para construir el AU, estos andamios son administrados en forma de código de conexión, para el *Component Manager*, junto con una jerarquía de clases de C++.

¹² APPLE INC. *Audio Unit Development Fundamentals*, *Audio Unit Programming Guide*. 2007.

La figura 5 representa el código personalizado como un cubo dentro del AU, y representa las clases del SDK, y el código de conexión como tiras conectando el cubo interno con el externo. Se puede desarrollar un AU sin utilizar el Core Audio SDK, pero esto involucraría tener que lidiar con más trabajo.

4.1.4.2 La Audio Unit Specification. La *Audio Unit Specification* define la interfaz común que el desarrollador de AU y los desarrolladores de aplicaciones *Host* deben tener.

La *AudioUnit Specification* describe:

- Todos los tipos de Apple definidos para Audio Units, como está listado en los "AudioUnit Components Types and Subtypes" enumerados en el archivo encabezador "AUComponent.h" en el almacén de archivos del AU.
- Los requerimientos funcionales y de comportamiento para cada tipo de AU.
- La *Plug-in API* para cada tipo de AU, incluyendo las propiedades opcionales y requeridas.

Se desarrolla un AU conforme a la *AudioUnit Specification*. Se puede probar esta "conformidad" con la herramienta de líneas de comando, descrita en la siguiente sección.

La *AudioUnit Specification* define la *Plug-in API* para los siguientes tipos de AU.

- Unidades de efectos ('aux'), como controles de volúmenes, ecualizadores, reverbs, que modifican el flujo de datos de audio.
- Unidades de efectos musicales ('aumf'), como secuenciadores, en los que se combina características de unidades de instrumentos (como reproducir y detener un sample) con características de unidades de efectos.
- Unidades de efectos Offline, las cuales permiten realizar procesos con el audio que no son posibles en tiempo real, como una inversión en el tiempo.
- Unidades de instrumento ('aumu'), las cuales toman MIDI y bancos de sonidos como una entrada y proveen datos de audio a la salida, permitiendo al usuario tocar un instrumento virtual.
- Unidades generadoras ('augn'), las cuales generan programáticamente una cadena de datos, o reproducen un archivo de audio.
- Unidades de conversión de datos ('aufc'), las cuales cambian las características de una trama de datos de audio como profundidad de bits, frecuencia de muestreo o velocidad de reproducción.

- Unidades de mezcla ('aumx'), las cuales combinan tramas de datos de audio.
- Unidades de paneo ('aupn'), las cuales distribuyen un set de canales de entrada utilizando un algoritmo de especialización, para determinar los canales de salida.

4.1.5 Funcionamiento del Audio Unit

4.1.5.1 Abrir y cerrar el Audio Units. Las aplicación anfitrionas son responsables, con ayuda del *Componet Manger*, de encontrar, abrir y cerrar los Audio Units; Los AU en respuesta, deben poderse encontrar, abrir y cerrar. El Audio Unit obtiene estos atributos cuando es construido con el *Core Audio SDK* y se utilizan las **plantillas de Xcode**.¹³

Hay una secuencia de dos pasos para que un *Audio Unit* esté disponible para el uso en el *Host*. Estos dos pasos son abrirlo e inicializarlo. Abrir un AU asciende a instancias de un objeto de la clase principal del AU; Inicializarlo asciende a localizar los recursos para que el AU esté listo para trabajar.

Para que el AU pueda comportarse bien, sea amigable con el anfitrión, la iniciación del AU debe ser rápida y liviana. El trabajo intensivo de arranque de los recursos para un AU, va en el proceso de inicialización; Por ejemplo un instrumento que implemente un banco grande de datos de muestras deberá cargarlo en su inicialización, no al ser llamado desde el programa anfitrión.

4.1.5.2 Inicialización múltiple. Un Audio Unit puede ser inicializado cualquier cantidad de veces deseadas por una aplicación anfitriona. Más precisamente, el *Componet Manager* invoca la inicialización del AU en nombre de la aplicación anfitriona. La infraestructura del *Componet Manager* asegura que la instancia de cada AU exista y se comporte independientemente.

4.1.5.3 Gráficos de procesamiento de Audio y el Modelo Pull (Arrastre). Aplicaciones anfitrionas se pueden conectar entre sí para que la salida de un *Audio Unit* alimente la entrada de la otra. Esta clase de series de Audio Units inter-conectados es llamada un "Gráfico de Procesamiento". En el contexto de un gráfico, cada Audio Unit conectado es un nodo.

La *Audio Processing Graph API*, declarada en la estructura de la Audio Toolbox, provee interfaces para asistir a aplicaciones anfitrionas a crear y manejar gráficos de procesamiento de audio. Cuando la aplicación anfitriona

13 APPLE INC. Audio Units in Action, Apple. Audio Unit Programming Guide, 2007.

emplea esta *API*, usa un tipo de datos opaco llamado un Objeto Gráfico (del tipo *AUGraph*).

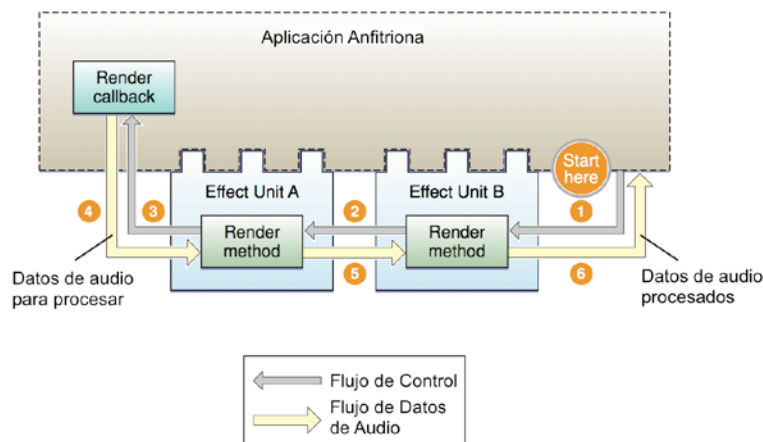
Algunas aplicaciones, como el *A.U. Lab*, siempre usan objetos gráficos para interconectar los *AU*. Otras como *Logic*, conectan los *AU* entre ellos directamente. Un *Audio Unit* individual, sin embargo, no está en conocimiento de donde la conexión es manejada por un Objeto Gráfico o por la aplicación anfitriona, o por el anfitrión directamente.

El flujo de datos de audio procede desde el primer hasta el último nodo (entrada y salidas correspondientemente), como sería de esperar. El control, sin embargo, fluye desde el último nodo de regreso al primero. En *Core Audio*, esto es llamado el Modelo *Pull* (Modelo de Arrastre). La aplicación anfitriona está encargada de invocar el arrastre.

Se puede pensar en el Modelo de Arrastre en términos de un pitillo de agua. El agua en el vidrio representa datos de audio frescos esperando a ser procesador. El pitillo representa un gráfico de procesamiento de audio, o incluso un solo *AU*. Actuando como una aplicación anfitriona, se comienza el flujo de datos de audio arrastrando (succionando) en el final del pitillo. Más específicamente, una aplicación anfitriona inicia el flujo de datos de audio llamando el método de *render* (impresión) del nodo final en un gráfico.

Antes de que comience el flujo de audio, la aplicación anfitriona realiza el trabajo de enganchar los *Audio Units* entre sí. En el caso de *AU Lab*, este hace conexiones hacia y desde el gráfico de procesamiento de audio. Pero los anfitriones no necesariamente alimentan los datos de audio a los gráficos que utilizan. En un caso donde el primer *AU* en un gráfico sea un generador no hay conexión de entrada, el generador provee datos automáticamente o reproduciendo un archivo.¹⁴

Figura 6. El Modelo *Pull* en acción con dos unidades de efectos



Audio Unit Programming Guide.

¹⁴ APPLE INC. Audio Units in Action, Audio Unit Programming Guide, 2007.

La aplicación *Host* llama al método de *render* del nodo final (unidad de efecto B) en el Gráfico, pidiendo por una porción de suficiente información de cuadros de datos de audio procesados.

El método de *render* de la unidad de efecto B busca en su buffer de entrada por datos de audio para procesar, para satisfacer la llamada a responder. Si hay datos de audio esperando para ser procesados, la unidad de efecto B los usa. De otra manera, como es mostrado en la figura 6, la unidad de efecto B (empleando una superclase en la jerarquía de clases de la SDK), llama al *método de render* de lo que sea que el *Host* tenga conectado a las entradas de la unidad de audio B. En este ejemplo, la unidad de efecto A esta conectada a las entradas de la unidad de efecto B, así que la unidad de efecto B jala a la unidad de efecto A, pidiendo por una tajada de cuadros de datos de audio.

La unidad de efecto A se comporta como la Unidad de efecto B. Cuando necesita datos de audio, los toma desde su conexión de entrada, la cual también está establecida por el *Host*. EL *Host* conecta las entradas de la unidad de efecto al *render callback* en el *Host*. La unidad de efecto A jala el *render callback* del *Host*.

El *render callback* del anfitrión provee las porciones de datos de audio a la unidad de efecto A.

La unidad de efecto A procesa la tajada de datos proveídos por el *Host*. La unidad de efecto A provee los cuadros de datos de audio procesados que habían sido previamente requeridos (paso 2) por la unidad B.

La unidad de efecto B procesa la tajada de datos proveída por la unidad de efecto A. Entonces la unidad de efecto B provee los cuadros de datos procesados que habían sido pedidos en primera instancia (paso 1) para la aplicación anfitriona. Esto completa el ciclo de Arrastre o Pull.

Un Audio Unit normalmente no sabe si sus entradas y salidas están conectadas a otro Audio Unit o a una aplicación anfitriona, o a algo más. Los Audio Units simplemente responden a las llamadas de *rendering*. Los anfitriones están a cargo de establecer las conexiones, y superclases (Para Audio Units construidos con SDK) encargándose de implementar el Jalado o *Pull*.

Un desarrollador no necesita trabajar directamente con los gráficos de procesamiento para asegurar que su Audio Unit funcione bien con ellos. Lo hace en parte asegurando que su Audio Unit pase el test de validación de Apple. También se puede hacer esta prueba enganchando el Audio Unit en varios gráficos de procesamiento utilizando aplicaciones *Host*.

4.1.6 Procesamiento. Un Audio Unit que procesa datos de audio, como una unidad de efecto, trabaja en términos de ciclos de *render* o entrega.¹⁵ En cada ciclo de *render*, el Audio Unit realiza los siguientes procesos:

- Toma una porción de cuadros de datos de audio frescos para procesar, esto lo hace llamando a la función de *render callback* que ha sido registrada en el Audio Unit.
- Procesa las porciones de datos de audio.
- Pone los datos de audio resultantes en los *buffers* de salida del Audio Unit.

Un Audio Unit realiza este trabajo a la "seña y llamada" de su aplicación *Host*. Esta aplicación también establece el número de cuadros de datos de audio por porción.

La llamada programática a entregar la siguiente porción de datos de audio puede llegar de los siguientes dos lugares:

- Desde la aplicación anfitriona, en el caso en que esta aplicación este utilizando el Audio Unit directamente.
- Desde el siguiente Audio Unit en la cadena, en el caso en que un Audio Unit este dentro de una gráfico de procesamiento

Los Audio Units se comportan de igual manera sin importar el contexto de la llamada, es decir si es llamado por el Host o por el siguiente Audio Unit en la cadena.

4.1.6.1 Reiniciar - Resetting. Los Audio Units también deben tener la habilidad de dejar de entregar datos. Por ejemplo, un Audio Unit implementa un filtro IIR usa un buffer interno de muestras. Usa los valores de las muestras en el Buffer cuando aplica una curva de frecuencia a las muestras procesadas. Si el usuario de este AU deja de reproducir un archivo de audio y comienza a reproducirlo nuevamente en un punto diferente, el AU en este caso debe empezar con el buffer de procesamiento vacío para evitar que se induzcan elementos anteriores.

Cuando se desarrolla el código de DSP para un AU, se debe implementar un método *Reset* para devolver el estado de DSP del AU a como estaba cuando el AU fue Inicializado en un principio. La aplicación anfitriona llama el método de *Reset* cuando lo necesita.

¹⁵ APPLE INC. Processing: The Heart of the Matter, Audio Unit Programming Guide, 2007.

4.1.6.2 Ajustes en el Render (Impresión). Cuando un AU imprime los datos, un usuario puede utilizar el comportamiento de impresión utilizando la vista del AU. Por ejemplo, en un AU de filtro paramétrico, un usuario puede ajustar la frecuencia central. También es posible para las aplicaciones *Host* de alterar la impresión utilizando la automatización de parámetros.

4.1.6.3 Automatización de Parámetros. Los parámetros permiten a los usuarios ajustar los AUs. Por ejemplo, El filtro *Low-pass* de Apple tiene parámetros para frecuencia de corte y resonancia. La automatización de parámetros permite a los usuarios ajustes de parámetros a través una línea de tiempo. Por ejemplo, un usuario podría querer utilizar un AU de filtro de bajas frecuencias para proveer un efecto parecido al pedal de *wah* de una guitarra. Con la automatización de parámetros, el usuario podría grabar el efecto de *wah* y hacerlo parte de una composición musical. La aplicación anfitriona graba los cambios manuales junto con la información de sincronía, escriben los cambios en los marcadores de audio en un canal de audio. El *Host* puede reproducir en regreso los cambios en los parámetros para proveer control automatizado del AU.

Las aplicaciones también pueden proveer la habilidad de que un usuario pueda indirectamente especificar la manipulación de parámetros. Por ejemplo, un *Host* puede permitir a un usuario dibujar una curva de paneo o ganancia a través de la representación en forma de onda en un canal de audio. El *Host* puede traducir esa entrada gráfica en datos de parámetros de automatización.

La automatización de parámetros depende de tres cosas:

- La habilidad de un AU de cambiar el valor de sus parámetros programáticamente en respuesta a una aplicación *Host*.
- La habilidad la vista de un AU de publicar notificaciones mientras los valores de los parámetros son cambiados por un usuario.
- La habilidad de la aplicación *Host* de soportar la grabación y reproducción de datos de automatización de parámetros.

Algunos *Host* que soportan automatización de parámetros con AUs son Logic Pro, Ableton Live, y Sagan Metro.

La automatización de parámetros usa la *Audio Unit Event API*, Declarada en el archivo encabezador *AudioUnitUtilities.h*, que es parte de la estructura del Audio Toolbox. Esta aplicación es de intercambio seguro, y provee un mecanismo de notificación que permite mantener los Audio Units, su vista y el *Hots* en sincronía.

Para proveer la automatización de parámetros en el Audio Unit, el programador debe crear una vista personalizada. Se provee el soporte para automatización en el código ejecutable de la vista, haciendo uso de la *Audio Unit Event API* para soportar todos o algunos de los siguientes tipos de eventos.

- Gestos de parámetros, que incluyen los tipos de eventos:
kAudioUnitEvent_BeginParameterChangeGesture,
kAudioUnitEvent_EndParameterChangeGesture.
- Cambios de valor en los parámetros, identificado por el tipo de evento:
kAudioUnitEvent_ParameterValueChange.
- Cambios de propiedades, identificados por el tipo de evento:
kAudioUnitEvent_PropertyChange

En algunos casos poco comunes el desarrollador podrá necesitar agregar soporte de automatización al *Audio Unit* en sí. Por ejemplo, se puede crear un filtro pasa bandas con frecuencias ajustables en los extremos inferior y superior. Este AU necesita entonces asegurar que la frecuencia de corte superior nunca pases por menos de la inferior. Cuando un AU invoca el cambio de parámetro en un caso como este, este necesita encargarse de realizar una notificación de cambio de parámetro.

4.2 MARCO LEGAL O NORMATIVO

- LEY NUMERO 23 DEL 28 DE ENERO DE 1982 SOBRE DERECHOS DE AUTOR. CONGRESO DE LA REPUBLICA DE COLOMBIA. DIRECCIÓN NACIONAL DEL DERECHO DE AUTOR. MINISTERIO DEL INTERIOR. Los autores de obras literarias, artísticas y científicas, como el desarrollo de software realizado en este proyecto, gozarán de protección para sus obras en la forma prescrita por la presente ley y, en cuanto fuere compatible con ella, por el derecho común.

- LA ORGANIZACIÓN MUNDIAL DE LA PROPIEDAD INTELECTUAL (OMPI). Este es un organismo especializado del Sistema de Naciones Unidas, creada en 1967 con la firma del Convenio de Estocolmo. OMPI está dedicada a fomentar el uso y la protección de las obras del intelecto humano, incluyendo el desarrollo de códigos para implementación de software, como el desarrollado en este proyecto de grado. Con sede en Ginebra (Suiza), OMPI es uno de los 16 organismos especializados del sistema de organizaciones de las Naciones Unidas.

- COPYLEFT. El *copyleft* es una práctica al ejercer el derecho de autor que consiste en permitir la libre distribución de copias y versiones modificadas de una obra u otro trabajo, exigiendo que los mismos derechos sean preservados en las versiones modificadas. La efectividad de ejercerlo puede depender de la legislación particular a cada país, pero en principio se puede utilizar para programas informáticos, obras de arte, cultura, ciencia, o cualquier tipo de obra o trabajo creativo que sea regido por el derecho de autor. XCode y SonicBirth dos softwares de desarrollo utilizados en este proyecto permiten ejercer este tipo de derecho.

5. DESARROLLO INGENIERIL

5.1 DEFINICIÓN DE LA PLATAFORMA DE PROGRAMACIÓN SOBRE LA CUAL SE DESARROLLARA EL PLUG-IN.

Para la implementación de este *AudioUnit* se analizaron previamente dos plataformas de desarrollo, Xcode Tools de Apple y SonicBirth, mediante el desarrollo de *Audio Units* de "poca complejidad". Se analizó el desarrollo de un "Tremolo Unit" en la plataforma **Xcode Tools** de Apple, según es descrito en la *Audio Unit Programming Guide* de esta misma compañía, y posteriormente se desarrolló un "Reverberador de Schroeder" en la plataforma **SonicBirth**, desarrollada como un proyecto para el École Polytechnique de Montreal.

Estos dos programas de desarrollo son de licencia gratuita para las personas que deseen utilizarlos, así mismo se encuentra códigos con diferentes ejemplo de desarrollo de *AudioUnits*.

5.1.1 Análisis del desarrolló de un "Tremolo Unit" en XcodeTools. Para el análisis de Xcode Tools se analizó el desarrolló un Audio Unit de tremolo sencillo, que se llamó "Tremolo Unit".

Xcode es un programa de desarrollo que utiliza un tipo de programación llamada "Programación Enfocada a Objetos" en código C++, dentro la API llamada **Carbon**, que utiliza la tecnología de la compañía Apple. El código se desarrolló principalmente sobre un archivo encabezador .h, que en este caso se llamó "TremoloUnit.h" y un archivo de implementación que se llamó "TremoloUnit.cpp".

El software XCode Tools es *Open Source*, y se puede descargar gratuitamente desde la página de Apple¹⁶, incluye varias aplicaciones para el desarrollo y prueba del *software* en *Carbon* para Macintosh, como el AU Lab que será utilizado posteriormente en la verificación del funcionamiento del *Doppler Effect*.

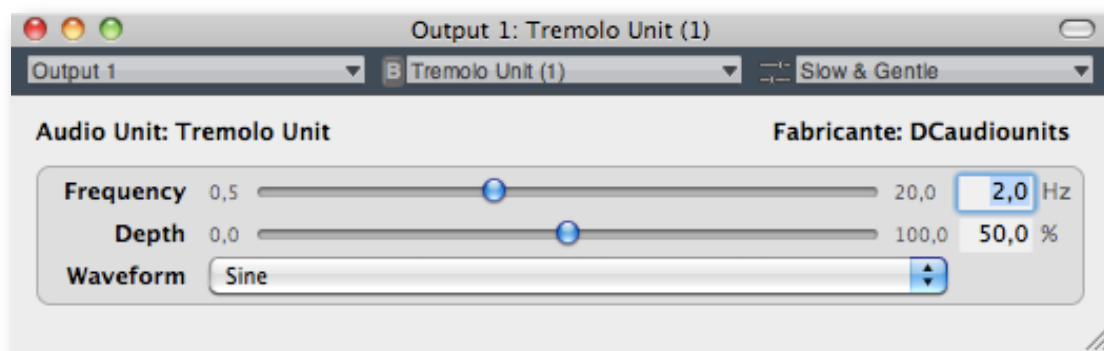
Se observó desarrollo el Tremolo Unit, según es descrito en la *Audio Unit Programming Guide* de Apple. Mediante la "Sub-clasificación" de las correctas "Super-clases C++" del *Core Audio SDK*, que también se puede descargar gratuitamente en la página de Apple.

Aunque las plantillas de Xcode Tools aíslan al desarrollador de realizar la mayor parte de las conexiones programáticas del *AudioUnit* con el programa *Host*, realizar el código DSP, y el código para la interfaz gráfica de usuario personalizada, es de gran complejidad para un Ingeniero de Sonido, sin conocimientos avanzados para este tipo de código programático, aun siguiendo la guía brindaba por Apple.

¹⁶ <http://developer.apple.com/tools/xcode/>

Para este *AudioUnit* se desarrolló una interfaz gráfica de usuario “genérica”, proveída por las plantillas de *AudioUnit* de XCode. No se desarrolló interfaz gráfica de usuario personalizada, ya que el desarrollo del código para esta es algo que no es concerniente con un Ingeniero de Sonido, más allá del diseño y verificación del funcionamiento del mismo.

Figura 7. Tremolo Unit desarrollado en XCode Tools (GUI)



Definición de la plataforma de programación.

5.1.2 Desarrollo de Reverberador de Schroeder en SonicBirth. La segunda plataforma que se analizó fue SonicBirth, mediante el desarrollo de un Audio Unit de “Reverberación de Schroeder”.

Esta aplicación fue desarrollada inicialmente como un proyecto para el École Polytechnique de Montreal, y posteriormente desarrollada como un producto completo.¹⁷

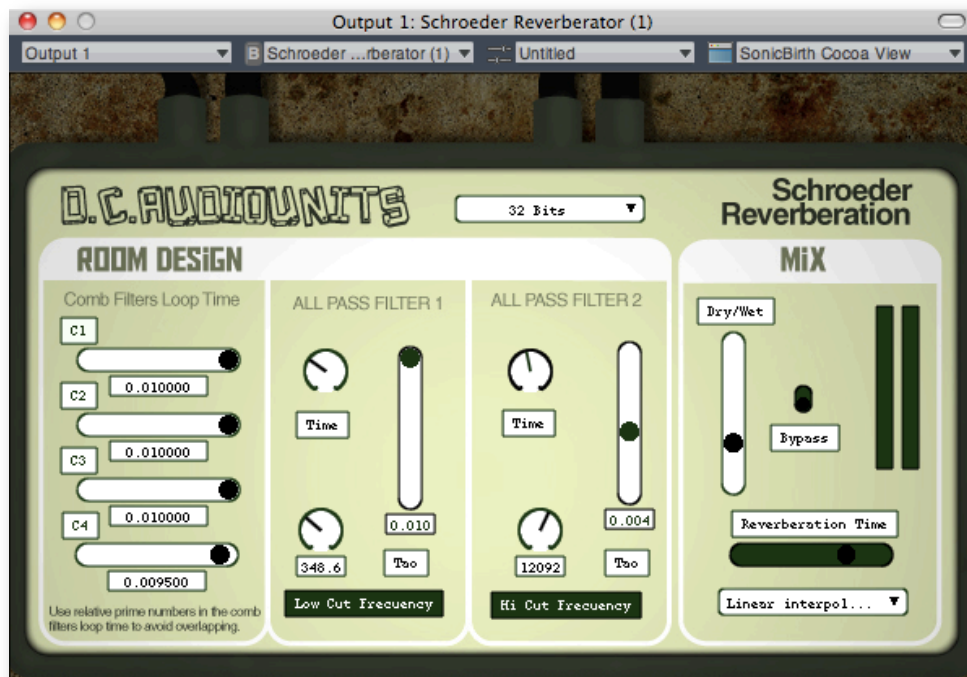
En este se implementa el DSP mediante la realización de un “Diagrama de Bloques”, los cuales son llamados circuito, y permite al usuario utilizar una serie de elementos programáticos para el desarrollo del código DSP, elementos algebraicos, trigonométricos, argumentos, generadores y analizadores, entre otros, rutiando los datos dados por el DAW entre ellos, para realizar el diagrama del código que implemente la función deseada.

En este, la interfaz de usuario es desarrollada de manera mucho más gráfica, permitiendo manipular la forma y ubicación de parámetros y elementos. Permite utilizar elementos gráficos como medidores de nivel y osciloscopios para en el GUI la información necesaria para la correcta manipulación del software. También permite controlar algunos parámetros utilizando control MIDI.

En el anexo 6 se muestra el diagrama desarrollado para esta aplicación.

¹⁷ <http://sonicbirth.sourceforge.net/>

Figura 8. Schroeder Reverberator desarrollado en SonicBirth (GUI)



Definición de la plataforma de programación.

5.1.3 Conclusiones y Comparación. En este punto del análisis, se concluye que XCode y SonicBirth son dos herramientas de desarrollo de *Audio Units* que trabajan de manera muy diferente, pero ambos permiten desarrollar *plug-ins* de muy buena calidad, con completa funcionalidad, que además permiten desarrollar una interfaz de usuario personalizada.

Existe un mayor grado de complejidad programático al desarrollar el Audio Unit en XCode Tools y es necesario tener bases en programación enfocada a objetos en C. SonicBirth, por ser una herramienta de desarrollo basada en diagramas de bloques, permite mayores opciones para un Ingeniero de Sonido, con bases en sonido, para desarrollar tanto el código del DSP, como la Interfaz Gráfica de Usuario personalizada.

Una segunda conclusión de este análisis, es que Xcode Tools permite agregar funciones que no permite SonicBirth, como la posibilidad de agregar un código de protección de copia, además de poder personalizar completamente la interfaz gráfica de usuario, cosa que SonicBirth no permite por completo.

En un principio un ingeniero de diseño preferiría utilizar la herramienta SonicBirth para diseño y prueba del algoritmo en SonicBirth, y en un segundo lugar el completo desarrollo del código C del Audio Unit en Xcode, para realizar una interfaz más personalizada, o para realizar un proceso que no haya podido ser realizado con el programa SonicBirth

5.2 DISEÑO DEL ALGORITMO PARA EFECTO DOPPLER

5.2.1 Documento de Diseño El *plug-in* Doppler Effect de DC AudioUnits, realizará una emulación digital del “Efecto Doppler” a una señal deseada, basándose en las formulas descritas por el físico Christian Doppler, dando una sensación de movimiento dentro del espacio acústico por medio de un *plug-in*, que podrá ser utilizado en formato Audio Unit para Logic, Garage Band, y otros DAWs.

5.2.2 Análisis de la fórmula para el Doppler Pitch Shift. Partiendo de la formula descrita por el físico Christian Doppler, realizar el siguiente análisis, asumiendo que el observador no se encuentra en movimiento.

Ecuación 15. Ecuación utilizada en el diseño del del *plug-in*

$$\begin{array}{c} \text{Si } V_o=0 \\ f' = \frac{V_s \cdot f}{V_s - V_e} \\ f' = \frac{342 \text{ m/s} \cdot f}{342 \text{ m/s} - V_e} \end{array}$$

La velocidad de la fuente se asume como positiva si la fuente se aleja del receptor y negativa si esta se acerca.

5.2.2.1 Análisis del cambio en la frecuencia de Doppler en relación a la velocidad del emisor, temperatura y frecuencia. En base al análisis descrito por el físico Christian Doppler se analiza el cambio en diferentes frecuencias, velocidades y temperaturas, las tres variables principales dentro de la fórmula, para poder relacionar estos resultados con el cambio que se intentara obtener en el *plug-in*, y emula el efecto Doppler los más precisamente posible.

Para esto se obtendrán los resultados a 3 km/h, 30 km/h, 60 km/h, 120 km/h y 180 km/h, la velocidad máxima que un automóvil promedio alcanza. Se asumirá una velocidad de emisor constante. La frecuencia Doppler será analizada en diferentes octavas que cubren casi todo el espectro de audición humana, desde 32 Hz a 16000 Hz, también para cada velocidad se encontró el porcentaje de cambio en pitch que experimentaría la fuente sonora y el equivalente a este intervalo en cents para así relacionar los resultados con los que proveerá el Audio Unit.

La velocidad de la fuente se asume como positiva si la fuente se aleja del receptor y negativa si esta se acerca.

Tabla 1. Frecuencia Doppler a 3 km/h. La fuente se acerca. Velocidad de la fuente positiva

Vel. Emisor	Ve (m/s)	Vs (m/s)	f	f'	Cambio %
3 km/h	0.83 m/s	341.11 m/s	32 Hz	32.08 Hz	0.24
	0.83 m/s	341.11 m/s	64 Hz	64.16 Hz	0.24
	0.83 m/s	341.11 m/s	125 Hz	125.31 Hz	0.24
	0.83 m/s	341.11 m/s	250 Hz	250.61 Hz	0.24
	0.83 m/s	341.11 m/s	500 Hz	501.22 Hz	0.24
	0.83 m/s	341.11 m/s	1,000 Hz	1,002.45 Hz	0.24
	0.83 m/s	341.11 m/s	2,000 Hz	2,004.9 Hz	0.24
	0.83 m/s	341.11 m/s	4,000 Hz	4,009.8 Hz	0.24
	0.83 m/s	341.11 m/s	8,000 Hz	8,019.59 Hz	0.24
	0.83 m/s	341.11 m/s	16,000 Hz	16,039.18 Hz	0.24
Vel Sonido	341.11 m/s				
Temp	16 °C				
CAMBIO CENTS.		4.23			

Análisis para el Doppler *pitch shift*

Tabla 2. Frecuencia Doppler a 3 km/h. La fuente se aleja. Velocidad de la fuente negativa

Vel. Emisor	Ve (m/s)	Vs (m/s)	f	f'	Cambio %
-3 km/h	-0.83 m/s	341.11 m/s	32 Hz	31.92 Hz	-0.24
	-0.83 m/s	341.11 m/s	64 Hz	63.84 Hz	-0.24
	-0.83 m/s	341.11 m/s	125 Hz	124.7 Hz	-0.24
	-0.83 m/s	341.11 m/s	250 Hz	249.39 Hz	-0.24
	-0.83 m/s	341.11 m/s	500 Hz	498.78 Hz	-0.24
	-0.83 m/s	341.11 m/s	1,000 Hz	997.56 Hz	-0.24
	-0.83 m/s	341.11 m/s	2,000 Hz	1,995.13 Hz	-0.24
	-0.83 m/s	341.11 m/s	4,000 Hz	3,990.25 Hz	-0.24
	-0.83 m/s	341.11 m/s	8,000 Hz	7,980.5 Hz	-0.24
	-0.83 m/s	341.11 m/s	16,000 Hz	15,961.01 Hz	-0.24
Vel Sonido	341.11 m/s				
Temp	16 °C				
CAMBIO CENTS.		-4.22			

Análisis para el Doppler *pitch shift*

Como se puede observar en la tabla 1, a 3 kilómetros por hora el intervalo del cambio Doppler es de menos del 1 %, se puede asumir que ha esta velocidad el efecto es imperceptible.

En la tabla 2, se muestra este cambio pero en dirección contraria, se puede ver que es el mismo en cents, en ambas direcciones, la diferencia es que al acercarse, la frecuencia es mayor y al alejarse esta disminuye. Más adelante se describe la fórmula utilizada para hallar este cambio en cents.

Esto confirma que a medida que la fuente se acerca se percibe una frecuencia mayor, pero va disminuyendo, cuando pasa por en frente del observador es la misma, y al alejarse de este sigue disminuyendo.

También podemos ver que el efecto es proporcional en cada octava, lo que quiere decir que aunque en 32 Hz tuvo un cambio de menos de 1 Hertz, en 16000 tiene un cambio en la misma proporción logarítmica, de más de 26 Hertz, y es percibido de igual manera en todo el espectro, por la percepción "logarítmica" del oído humano.

Tabla 3. Frecuencia de Doppler a 30km/h

Vel. Emisor	Ve (m/s)	Vs (m/s)	f	f'	Cambio %
30 km/h	8.33 m/s	341.11 m/s	32 Hz	32.8 Hz	2.50
	8.33 m/s	341.11 m/s	64 Hz	65.6 Hz	2.50
	8.33 m/s	341.11 m/s	125 Hz	128.13 Hz	2.50
	8.33 m/s	341.11 m/s	250 Hz	256.26 Hz	2.50
	8.33 m/s	341.11 m/s	500 Hz	512.52 Hz	2.50
	8.33 m/s	341.11 m/s	1,000 Hz	1,025.04 Hz	2.50
	8.33 m/s	341.11 m/s	2,000 Hz	2,050.08 Hz	2.50
	8.33 m/s	341.11 m/s	4,000 Hz	4,100.17 Hz	2.50
	8.33 m/s	341.11 m/s	8,000 Hz	8,200.33 Hz	2.50
	8.33 m/s	341.11 m/s	16,000 Hz	16,400.67 Hz	2.50
Vel Sonido	341.11 m/s				
Temp	16 °C				
CAMBIO CENTS.		42.82			

Análisis para el Doppler *pitch shift*

Al aumentar la velocidad 10 veces, es decir a 30 kilómetros por hora, el cambio Doppler no es mayor al 3%, es decir que el cambio de tono no es directamente proporcional a la velocidad. También se puede apreciar el valor del cambio Doppler en cents.

Tabla 4. Frecuencia Doppler A 60 Km/h

Vel. Emisor	Ve (m/s)	Vs (m/s)	f	f'	Cambio %
60 km/h	16.67 m/s	341.11 m/s	32 Hz	33.64 Hz	5.14
	16.67 m/s	341.11 m/s	64 Hz	67.29 Hz	5.14
	16.67 m/s	341.11 m/s	125 Hz	131.42 Hz	5.14
	16.67 m/s	341.11 m/s	250 Hz	262.84 Hz	5.14
	16.67 m/s	341.11 m/s	500 Hz	525.68 Hz	5.14
	16.67 m/s	341.11 m/s	1,000 Hz	1,051.37 Hz	5.14
	16.67 m/s	341.11 m/s	2,000 Hz	2,102.74 Hz	5.14
	16.67 m/s	341.11 m/s	4,000 Hz	4,205.48 Hz	5.14
	16.67 m/s	341.11 m/s	8,000 Hz	8,410.96 Hz	5.14
	16.67 m/s	341.11 m/s	16,000 Hz	16,821.92 Hz	5.14
Vel Sonido	341.11 m/s				
Temp	16 °C				
CAMBIO CENTS.		86.72			

Análisis para el Doppler *pitch shift*

Tabla 5. Frecuencia Doppler a 120 km/h

Vel. Emisor	Ve (m/s)	Vs (m/s)	f	f'	Cambio %
120 km/h	33.33 m/s	341.11 m/s	32 Hz	35.47 Hz	10.83
	33.33 m/s	341.11 m/s	64 Hz	70.93 Hz	10.83
	33.33 m/s	341.11 m/s	125 Hz	138.54 Hz	10.83
	33.33 m/s	341.11 m/s	250 Hz	277.08 Hz	10.83
	33.33 m/s	341.11 m/s	500 Hz	554.15 Hz	10.83
	33.33 m/s	341.11 m/s	1,000 Hz	1,108.3 Hz	10.83
	33.33 m/s	341.11 m/s	2,000 Hz	2,216.61 Hz	10.83
	33.33 m/s	341.11 m/s	4,000 Hz	4,433.21 Hz	10.83
	33.33 m/s	341.11 m/s	8,000 Hz	8,866.42 Hz	10.83
	33.33 m/s	341.11 m/s	16,000 Hz	17,732.85 Hz	10.83
Vel Sonido	341.11 m/s				
Temp	16 °C				
CAMBIO CENTS.		178.02			

Análisis para el Doppler *pitch shift*

Tabla 6. Frecuencia Doppler A 180 km/h

Vel. Emisor	Ve (m/s)	Vs (m/s)	f	f'	Cambio %
180 km/h	50 m/s	341.11 m/s	32 Hz	37.5 Hz	17.18
	50 m/s	341.11 m/s	64 Hz	74.99 Hz	17.18
	50 m/s	341.11 m/s	125 Hz	146.47 Hz	17.18
	50 m/s	341.11 m/s	250 Hz	292.94 Hz	17.18
	50 m/s	341.11 m/s	500 Hz	585.88 Hz	17.18
	50 m/s	341.11 m/s	1,000 Hz	1,171.76 Hz	17.18
	50 m/s	341.11 m/s	2,000 Hz	2,343.51 Hz	17.18
	50 m/s	341.11 m/s	4,000 Hz	4,687.02 Hz	17.18
	50 m/s	341.11 m/s	8,000 Hz	9,374.04 Hz	17.18
	50 m/s	341.11 m/s	16,000 Hz	18,748.08 Hz	17.18
Vel Sonido	341.11 m/s				
Temp	16 °C				
CAMBIO CENTS.		274.41			

Análisis para el Doppler *pitch shift*

Se puede concluir que para una fuente que se desplaza dentro de las velocidades promedio de un automóvil, máximo a 180 km/h tendrá un cambio máximo de 12 % en la frecuencia fundamental. O de 274.41 Cents lo que equivale a más de 2 tonos y medio en la escala tonal cromática pitagórica de occidente.

También se puede observar que al duplicar la velocidad, la frecuencia de Doppler aumenta casi en la misma relación,

5.2.2.2 Cambio en cents de la frecuencia Doppler. Para establecer una relación entre el cambio en frecuencia con un cambio en cents, entre los datos obtenidos utilizando la fórmula y los datos que proveerá el Audio Unit de Doppler, se haya un valor para el intervalo del cambio en Cents utilizando la siguiente fórmula, se utilizó la frecuencia *a* como la original y *b* la frecuencia resultante al ser afectada la señal al efecto Doppler¹⁸:

Ecuación 16. Fórmula intervalo en cents entre frecuencia *a* y *b*.

$$n = 1200 \cdot \log_2 \left(\frac{b}{a} \right) \approx 3986 \cdot \log_{10} \left(\frac{b}{a} \right)$$

¹⁸ ELLIS, Alexander . On the Musical Scales of Various Nations. 1885.

La tabla 7 muestra que el cambio de Doppler es el mismo en todas las frecuencias, midiendo este cambio en Cents, lo que quiere decir que se percibe de igual manera en todo el espectro. En la siguiente tabla se muestra un ejemplo de un cambio Doppler a una velocidad de desplazamiento de fuente de 100 kilómetros por hora con una temperatura de 16 grados centígrados. Fue calculada según la fórmula de Christian Doppler.

Tabla 7. Cambio Doppler en Cents a 100km/h, en todo el espectro.

FRECUENCIA	FREC. DOPPLER	CAMBIO CENTS
32 Hz	34.84 Hz	147.05
64 Hz	69.67 Hz	147.05
125 Hz	136.08 Hz	147.05
250 Hz	272.16 Hz	147.05
500 Hz	544.33 Hz	147.05
1,000 Hz	1,088.65 Hz	147.05
2,000 Hz	2,177.3 Hz	147.05
4,000 Hz	4,354.61 Hz	147.05
8,000 Hz	8,709.22 Hz	147.05
16,000 Hz	17,418.44 Hz	147.05

Cálculos según fórmula de efecto Doppler.

El siguiente paso es analizar la manera en que la temperatura, la cual provoca una variación en la velocidad del sonido, afecta la frecuencia obtenida por el efecto Doppler. Aquí se observa que al aumentar la temperatura disminuye el efecto Doppler en una relación pequeña, y más aún en bajas velocidades del emisor. Por ejemplo, a 100 kilómetros por hora se disminuye únicamente 5 Cents al duplicar la temperatura, esta es la factor máxima en que la temperatura afecta al efecto producido por el desplazamiento de la fuente.

Esto quiere decir que un cambio en la temperatura no afecta significativamente el cambio Doppler, ya que un intervalo de 5 Cents es imperceptible para el oído humano.

La siguiente tabla muestra el cambio Doppler a diferentes temperaturas, con una misma velocidad de desplazamiento de la fuente, fue calculada según la fórmula de Christian Doppler.

Tabla 8. Cambio Doppler en Cents, a diferentes temperaturas

Temperatura	100km/h	50km/h	25km/h
16 °C	147.05 Cents	71.97 Cents	35.61 Cents
18 °C	146.51 Cents	71.7 Cents	35.48 Cents
24 °C	144.9 Cents	70.93 Cents	35.1 Cents
26 °C	144.37 Cents	70.68 Cents	34.98 Cents
32 °C	142.81 Cents	69.93 Cents	34.61 Cents
36 °C	141.79 Cents	69.44 Cents	34.37 Cents

Cálculos según fórmula de efecto Doppler.

5.3 DISEÑO DE LOS PARÁMETROS DE LA INTERFAZ DE USUARIO (GUI)

Acá se especifican los parámetros con los que el usuario manipulará las funciones en el Audio Unit y los cuales vera en la interfaz gráfica de usuario (GUI). Se nombra cada uno, y se describe la manera en que trabajará, pensando en el diseño del Audio Unit.

- Posición de la fuente. (Pad X,Y)
- Velocidad de desplazamiento de la fuente en dB. (Fader)
- Activación del oscilador para el desplazamiento vertical (Switch)
- Velocidad de desplazamiento de la fuente en paneo. (Fader)
- Activación del oscilador para el desplazamiento de la fuente en paneo. (Switch)
- Forma de onda de la oscilación. (Senoidal, Diente de Sierra, Triangular)
- Doppler Pitch Shift. (on, off), (Fader)

5.3.1 Definición de los parámetros y sus *presets*. Se definen las funciones y valores de los parámetros nombrados anteriormente, especificando su nombre en la interfaz para el usuario, un nombre programático que se utiliza para el desarrollo del algoritmo en XCode, la unidad en que se mide el parámetro, y se especifica su valor máximo y mínimo, para así tener un acercamiento más claro al diseño del Audio Unit.

También se da una descripción de la función que cada parámetro desarrollará dentro del Audio Unit.

Tabla 9. Descripción de los parámetros del Audio Unit

ATRIBUTO	VALOR	ATRIBUTO	VALOR
Nombre interfaz de usuario	Posición de la fuente (X,Y)	Nombre interfaz de usuario	Vel Osc 1. (Nivel)
Descripción	Define la posición de la fuente en el estéreo, y el nivel de la fuente.	Descripción	Definirá la velocidad del oscilador en nivel sonoro.
Nombre Programático para XCode	kDoppler_posf	Nombre Programático para XCode	kDoppler_veln
Unidad de Medición	En Y db (0,100) y en X paneo (-1,1)	Unidad de Medición	Hz
Valor Mínimo	X0, Y0	Valor Mínimo	5
Valor Máximo	X1, Y1	Valor Máximo	0
Valor Default	X1, Y0	Valor Default	5

ATRIBUTO	VALOR	ATRIBUTO	VALOR
Nombre interfaz de usuario	Osc 1 (On/Off)	Nombre interfaz de usuario	Forma de Onda Osc 1
Descripción	Activará y desactivará el oscilador de nivel	Descripción	Determinará la forma de desplazamiento de la fuente
Nombre Programático para XCode	kDoppler_swin	Nombre Programático para XCode	kDoppler_onda1
Unidad de Medición	On, Off	Unidad de Medición	Indexada
Valor Mínimo	Off, 0	Valor Mínimo	Senoidal
Valor Máximo	On, 1	Valor Máximo	Triangular
Valor Default	Off, 0	Otro Valor	Diente de Sierra
		Valor Default	Triangular

Diseño del *Audio Unit* de efecto Doppler

Tabla 9. Descripción de los parámetros del Audio Unit (Continuación).

ATRIBUTO	VALOR
Nombre interfaz de usuario	Osc 1 Sync (on/off)
Descripción	Permite activar o desactivar la opción de sincronización en la velocidad del Oscilador de Nivel.
Nombre Programático para XCode	kDoppler_ond1s
Unidad de Medición	On, Off
Valor Mínimo	Off, 0
Valor Máximo	On, 1
Otro Valor	--
Valor Default	Off

ATRIBUTO	VALOR
Nombre interfaz de usuario	Vel Sync Osc 1
Descripción	Permite elegir la velocidad sincronizada en el oscilador de nivel en valores musicales.
Nombre Programático para XCode	kDoppler_ond1v
Unidad de Medición	Tiempos musicales.
Valor Mínimo	Redonda
Valor Máximo	Negra
Otro Valor	Tresillo, Blanca
Valor Default	Negra

ATRIBUTO	VALOR
Nombre interfaz de usuario	Forma de Onda Osc 2
Descripción	Determinará la forma onda del oscilador de desplazamiento de la fuente.
Nombre Programático para XCode	kDoppler_onda2
Unidad de Medición	Indexada
Valor Mínimo	Seno
Valor Máximo	Triangular
Otro Valor	Diente de Sierra
Valor Default	Triangular

ATRIBUTO	VALOR
Nombre interfaz de usuario	Osc 2 (On/Off)
Descripción	Activará y desactivará el oscilador de nivel
Nombre Programático para XCode	kDoppler_swip
Unidad de Medición	On, Off
Valor Mínimo	Off, 0
Valor Máximo	On, 1
Valor Default	Off, 0

ATRIBUTO	VALOR
Nombre interfaz de usuario	Vel Osc 2 (Paneo)
Descripción	Definirá la velocidad del oscilador que manipula la onda entre el canal izquierdo y derecho del Estéreo.
Nombre Programático para XCode	kDoppler_velp
Unidad de Medición	Hz
Valor Mínimo	5
Valor Máximo	0
Valor Default	5

ATRIBUTO	VALOR
Nombre interfaz de usuario	Vel Sync Osc 2
Descripción	Permite elegir la velocidad sincronizada en el oscilador de nivel en valores musicales.
Nombre Programático para XCode	kDoppler_ond2v
Unidad de Medición	Tiempos musicales.
Valor Mínimo	Redonda
Valor Máximo	Negra
Otro Valor	Tresillo, Blanca
Valor Default	Negra

Diseño del *Audio Unit* de efecto Doppler

Tabla 9. Descripción de los parámetros del Audio Unit (Continuación).

ATRIBUTO	VALOR	ATRIBUTO	VALOR
Nombre interfaz de usuario	Osc 1 Sync (on/off)	Nombre interfaz de usuario	Doppler Pitch Shift (On/Off)
Descripción	Permite activar o desactivar la opción de sincronización en la velocidad del Oscilador de Paneo.	Descripción	Activara la opcion del cambio Doppler.
Nombre Programático para XCode	kDoppler_ond2s	Nombre Programático para XCode	kDoppler_doppler
Unidad de Medición	On, Off	Unidad de Medición	On/Off
Valor Mínimo	Off, 0	Valor Mínimo	Off, 0
Valor Máximo	On, 1	Valor Máximo	On, 1
Otro Valor	--	Valor Default	Off
Valor Default	Off		

ATRIBUTO	VALOR
Nombre interfaz de usuario	Doppler Amount
Descripción	Definirá la cantidad de efecto percibida.
Nombre Programático para XCode	kDoppler_doppam
Unidad de Medición	Porcentaje
Valor Mínimo	0
Valor Máximo	100
Valor Default	100

Diseño del *Audio Unit* de efecto Doppler.

5.4 DEFINICIÓN DEL TIPO DEL AUDIO UNIT

Por ser un software que emulara un efecto acústico que modifica las características de la señal, el Audio Unit será del tipo `afx`.

Programáticamente se define de la siguiente manera: `kAudioUnitType_Effect: 'afx'`

5.5 INFORMACIÓN DEL BUNDLE DEL AUDIO UNIT. Información programática, acerca del "Bundle" del Audio Unit, necesaria para el bundle de archivos del AU:

Nombre del A.U. En Ingles: DC Doppler Effect

Nombre programático del AU: `Dcdoppler`

Versión del Bundle: 1.1.1.

Descripción Breve del Audio Unit: "Dcdoppler version 1.0, copyright © 2009, DC. Audio"

Sub-tipo de AU: 'dopl'

Nombre de la compañía: DC Audio

Nombre programático de la compañía: dcaudiounits

Nombre de la compañía en código de 4 caracteres: dcau

Nombre de dominio inverso: com.

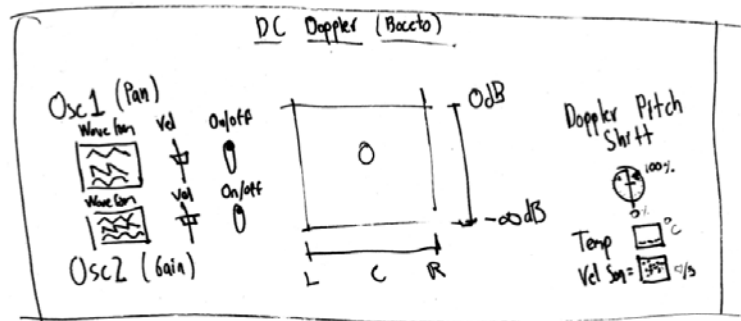
5.6 CREACION DEL "CÓDIGO DE DESARROLLADOR".

Apple recomienda que todos los desarrolladores registre un "Código de Desarrollador" en su página, que se utiliza en el desarrollo del Audio Unit.

- Decimal
68 67 97 117
- Hexadecimal
D C a u

5.7 BOCETO DEL AUDIO UNIT

Figura 9 Boceto de Doppler Effect



DC Doppler, DC AudioUnits

Los principales elementos involucrados en el efecto Doppler son la velocidad de la fuente, el cambio de tono percibido por el observador en relación a el tono original, y el nivel de presión sonora, que aumenta y disminuye a medida que la fuente se acerca y aleja del escucha. Para controlar estos elementos por medio de los para metros del *plug-in*, y emular el movimiento de la fuente se definieron los parámetros previamente mencionados.

El parámetro más notable en la interfaz de usuario será un PAD XY, con este definiría una posición central alrededor de la cual la fuente se desplazará, es un equivalente a la posición del escucha, alrededor del cual oscilará la fuente. Esta posición está dada en dos ejes, en el eje X se determina la

posición entre izquierda y derecha, en los canales del estereo de la fuente, en el eje Y se determina el nivel máximo que tendrá la fuente.

En segundo lugar para emular el desplazamiento de la fuente al acercarse y alejarse directamente al observador o de izquierda a derecha, se desarrollarán dos osciladores que definirán la trayectoria y velocidad a la que se desplazará la fuente sonora. La velocidad de los osciladores está relacionada con la velocidad de la fuente, por lo tanto entre más velocidad se le dé en algún eje mayor será el cambio Doppler percibido, siguiendo la teoría básica de efecto Doppler.

5.8 DESARROLLO DEL ALGORITMO PARA EFECTO DOPPLER, EN SONICBIRTH.

5.8.1 Desarrollo del Audio Unit de Paneo y Ganancia. En primer lugar se comenzará desarrollando un Audio Unit tipo "*Paner*" poder manipular la señal en los canales izquierdo y derecho del estéreo.

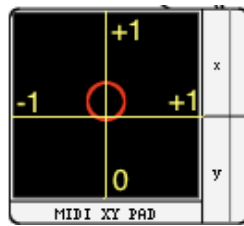
La herramienta SonicBirth funciona mediante el desarrollo de diagramas de bloques. A continuación se describirán los elementos programáticos que se utilizaron para el *audio unit* de Doppler Effect. En SonicBirth estos diagramas de bloques son llamados circuítos, y de esta manera serán nombrados durante el desarrollo del *audio unit*.

El movimiento que describirá la fuente será tanto de izquierda a derecha como en nivel para simular que la fuente se acerca y/o aleja, a este *paner* se le agregaran LFO (osciladores de baja frecuencia) para describir un movimiento, obviamente oscilatorio, ya sea de izquierda a derecha de lejos a cerca, o una mezcla de estos dos movimientos, estos osciladores describirán ondas senoidales y triangulares para permitir varios tipos de movimiento. El *Audio Unit* será estereofónico.

5.8.1.1 Pad MIDI XY. Con este definiría una posición central alrededor de la cual la fuente se desplazará, es un equivalente a la posición del escucha, alrededor del cual oscilará la fuente.

SonicBirth, posee un elemento, o bloque programático llamado "Pad MIDI XY", que permite manipular dos variables, Y en el eje vertical y en el horizontal X, con una esfera dentro del PAD que determina el valor de las variables. Para el Audio Unit de efecto Doppler la variable vertical tiene valores entre 0 y 1, y en la horizontal valores entre 1 y -1, y podrán ser controlados por medio del protocolo MIDI. Utilizando estas variables se manipularán el nivel de la señal, y la posición en estéreo, ver figura 10.

Figura 10. Pad MIDI XY



DC Doppler, SonicBirth

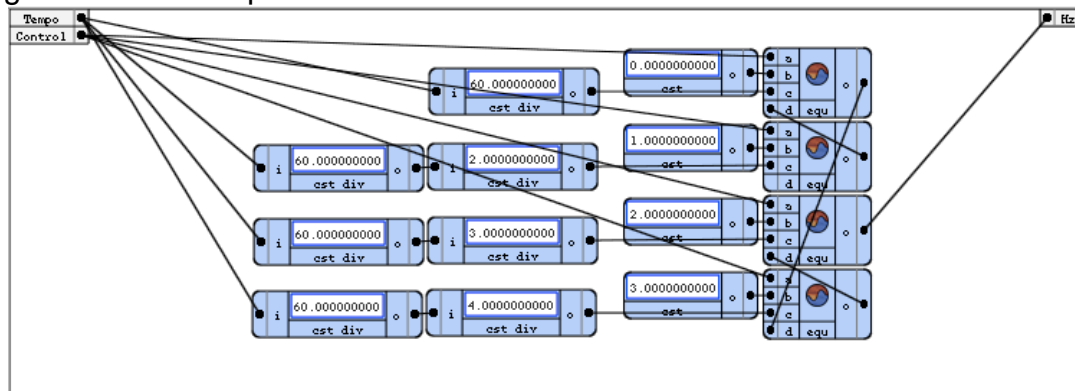
5.8.1.2 Osciladores de Nivel. EL oscilador de nivel definirá la velocidad y la forma de la trayectoria de la fuente sonora en el eje Y, es decir simula que la fuente se acerca y aleja del observador.

Para el desarrollo del oscilador de nivel, se utilizaron elementos programáticos, para multiplicar cada uno de los canales estéreo de la señal original por un valor que oscila en 0 y 1. Este valor define por el valor que el usuario ingrese en el PAD XY, en este caso el nivel es controlado desde el eje Y. O también es definido utilizando los osciladores.

Para este oscilador se desarrolló un sub-circuito que permite la opción de elegir entre una oscilación senoidal, triangular, o diente de sierra. El valor de las ondas oscila entre 0 y 1, no entre -1 y 1 como lo haría una senoidal o diente de sierra normal, ya que esta función determinará el nivel de la fuente, y no puede tener valores de nivel negativos. También se insertó un osciloscopio a la salida del generador de la onda, para que el usuario pueda observar más gráficamente la forma de onda del oscilador.

El usuario podrá manipular la velocidad del oscilador de nivel de dos maneras según lo desee, mediante un *Fader* que maneja valores entre 0.1 y 3 ciclos por segundo (Herz), o también podrá elegir en un menú, cuatro velocidades que estarán sincronizadas con el tempo de la sesión que se está trabajando en el DAW, las tres velocidades serán: negra, tresillo, blanca y redonda, utilizados en los tiempos rítmicos de la música occidental. Ver figura 11.

Figura 11. Circuito para velocidades sincronizadas



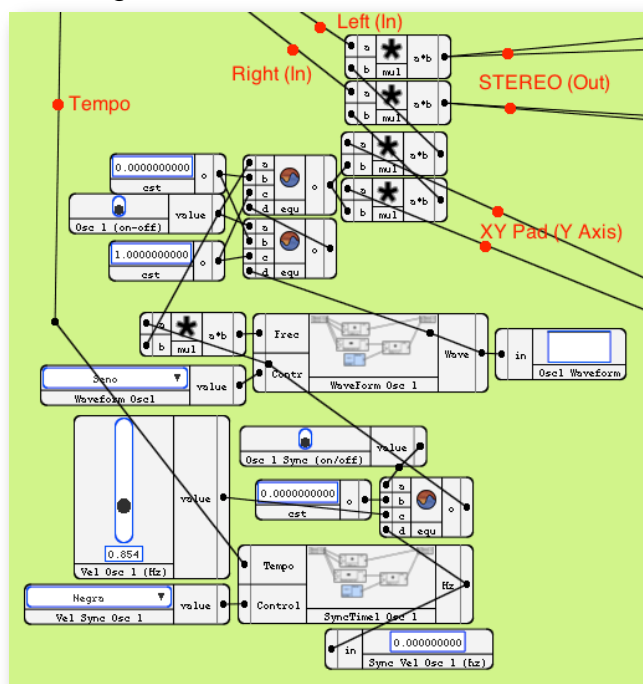
DC Doppler, SonicBirth

Este menú de velocidades sincronizadas fueron desarrollados utilizando comparadores para seleccionar el generador que emite la forma de onda que se desea usar, todos los osciladores se encontraran activos permanentemente, en total ocho para las formas de onda utilizadas en todo el circuito del efecto Doppler. Se utilizan dos diagramas idénticos para sincronizar los osciladores de nivel y paneo con el tiempo del programa anfitrión.

Además se desarrolló un circuito para utilizar velocidades sincronizadas, en tiempos musicales como negra, redonda, blanca, y tresillo. Al tomar el tiempo de la canción, este se divide por 60, que sería los segundos que contiene un minuto, así se llega a la velocidad que valdría una "negra", por ejemplo si el BPM de la canción son 120 al dividir este valor entre 60 la oscilación estaría en 2 Hz, o ciclos por segundo, al duplicar este valor se daría con la velocidad necesaria para la figura musical "blanca", y al cuadruplicarlo el valor para la "redonda".

Para los osciladores de nivel fue necesario multiplicar la señal de los generadores de onda de las señales senoidal y triangular por 0.5 y a esto sumarle 0.5, de esta manera se logra que la señal oscile entre 0 y 1. Si es desactivada la oscilación, el Pad X Y controlará el nivel. Ver figura 12.

Figura 12. Oscilador de ganancia



DC Doppler, SonicBirth

5.8.1.3 Paner y Oscilador de Paneo. Para emular un movimiento de la fuente izquierda a derecha del receptor y definir una posición central en la que esta se moverá se desarrolló un oscilador de baja frecuencia, que determinado por la

posición en el eje X del *Pad XY* describe un determinado movimiento en este eje, obviamente produciendo un cambio Doppler al alejar y acercarse del eje.

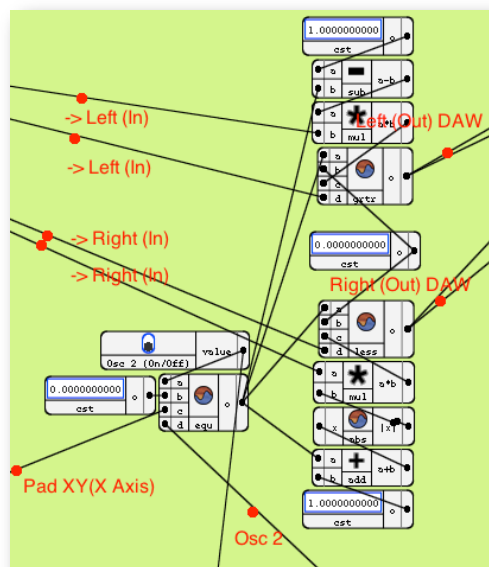
El *paneo*, o posición de la fuente entre los canales izquierdo y derecho del estéreo, podrá ser definido por el usuario ingresando un valor en el *Pad XY*, en el eje X, que oscila entre 1 para el canal Derecho y -1 para el canal Izquierdo. El *paner* se desarrolló por medio de comparadores, que permiten controlar el nivel entre izquierda y derecha imagen como se muestra en la figura 13.

Para crear un movimiento oscilador entre izquierda y derecha se añadieron generadores que determinan la posición al oscilar entre +1 y -1. Para emular diferentes trayectorias se desarrolló un "menú" en *SonicBirth*, por medio de elementos comparadores y variables indexadas, habilitando la opción de un movimiento oscilatorio con formas senoidal y triangular, o un movimiento en una sola dirección.

La forma de onda podrá ser elegida por el usuario en un menú que le permitirá elegir entre senoidal triangular o en movimiento tipo "One Shot". La frecuencia de estos osciladores también podrá ser controlada en Hertz, por un *Slider* que maneja valores entre 0.1 y 1.1 Hz, o por un menú que permite controlar velocidades sincronizadas con el Host, con las opciones de redonda, blanca, negra, y tresillo. En el figura se muestra el circuito utilizado para el oscilador de *paneo*.

Se desarrolló un control que manipula la amplitud de la oscilación entre izquierda y derecha, multiplicando la señal del generador por un factor entre cero y uno, también se desarrolló un diagrama de bloques para permitir al usuario elegir con el *PAD XY* la posición central en la que la fuente oscilará.

Figura 13. Circuito del *Paner*.



DC Doppler, *SonicBirth*

5.8.2 Pitch Shift. Finalmente para producir el cambio en pitch o tono, de la señal deseado que esta vas desde 4 cents a 3 Km/h, hasta 274 cents a 180 Km/h según lo analizado anteriormente sobre el Efecto Doppler.

Se analizaron varios métodos para producir un cambio en la tono, o pitch de la señal, el primero fue el análisis del paper “Fundamentos para *Pitch Shifting*”, Presentado en la convención 85 de la *Audio Engineering Society*, 1988 Noviembre 3-6, en Los Ángeles por Andy Duncan, Dave Rossum.

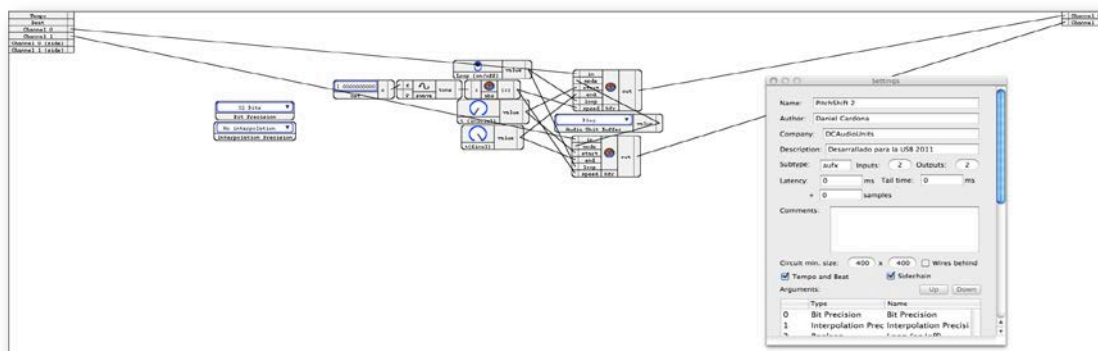
5.8.2.1 Pitch Shift 1. Además del circuito anterior, se desarrolló un segundo circuito de cambio de pitch, utilizando otro elemento programático dentro de SonicBirth llamado “*Buffersizer*”.

Este objeto programático permite almacenar un archivo de audio en un *Buffer* dentro del Audio Unit. Funciona en tres modos, *silence* (0), *play* (1), y *record* (2). En el modo *play*, los valores de *start* y *end* representan el desplazamiento de reproducción. Si *end* es menor que *start*, la velocidad es negada. Ambos valores deben estar entre 0 y 1.

La reproducción entra en un *loop*, o ciclo, si la entrada de este parámetro es diferente de 0. El parámetro Speed determina la velocidad de reproducción, y debe estar entre -5 y 5. Una velocidad negada significa que el archivo se reproducirá en reversa. En el modo *record*, el *buffer* se llenará desde el inicio hasta la capacidad que sea establecida, en este caso 10 segundos. EL problema sería que al bajar la velocidad cambia el tono, pero también es afectada la longitud del archivo, cambiando no solo el pitch si no también la velocidad de reproducción, lo cual no sería apropiado para emular correctamente el efecto Doppler.

A continuación se muestra el circuito para PitchShift 1.

Figura 14. Diagrama “Pitch Shift ”1



DC Doppler, SonicBirth

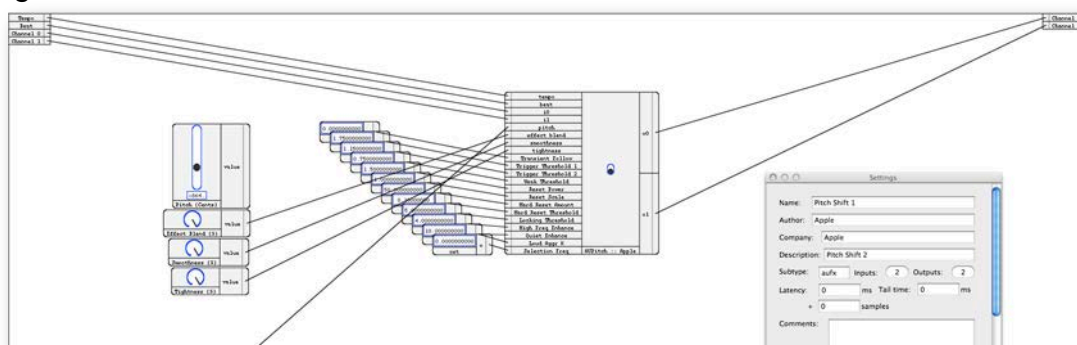
5.8.2.2 Pitch Shift 2. El primer método que se encontró para el cambio de tono, o pitch, fue obtenido mediante el uso de un bloque programático de SonicBirth llamado, "Third Party Audio Unit". Este bloque permite insertar dentro del Audio Unit que se está programando otro Audio Unit, de otro programador, que realice una función determinada.

En este caso para el cambio de tono, se utilizó un Audio Unit llamado AUPitch, desarrollado por Apple, y que se incluye en Garage Band, Logic, y cualquier programa de audio desarrollado por Apple.

Se establecieron valores fijos para la mayoría de los parámetros, y se desarrollaron controles sencillos para manipular el efecto, aunque posteriormente se utilizaran los osciladores para automatizar el cambio de tono en relación a las fórmulas de Doppler. Este Audio Unit permite cambiar el tono original del archivo de audio en "cents". Finalmente este fue el elemento utilizado para el cambio Doppler en el diagrama del *Audio Unit*.

En la figura 15 se muestra el circuito desarrollado, y los parámetros utilizados

Figura 15. Circuito "Pitch Shift 2"



DC Doppler, SonicBirth

5.8.3 Relación entre Velocidad, Nivel y Pitch de la señal para el efecto Doppler. Para emular el Efecto Doppler, se relaciona el nivel de la señal con la distancia. Como se vio en el análisis del efecto Doppler, si la fuente se acerca, se percibirá un tono superior al original, aunque este ira en decremento, y el nivel de presión sonora que percibe el receptor. Vendrá desde 0 hasta un nivel máximo el cual el receptor percibirá cuando la fuente este justo en frente de él. Al estar justo en frente del observador, el tono y el nivel de presión percibido por este será igual al nivel y tonos originales de la fuente.

Se concluyó que $-\infty$ dBfs, equivaldrían al nivel percibido por el receptor cuando la fuente se encuentra a una distancia máxima es decir, cuando la fuente que "se empieza acercar" y aún no se percibe, o cuando esta se ha alejado por completo, y 0dbfs, como el nivel que tendría la fuente en el momento en que esta se encuentra justo en frente del observador. Entonces una oscilación en el

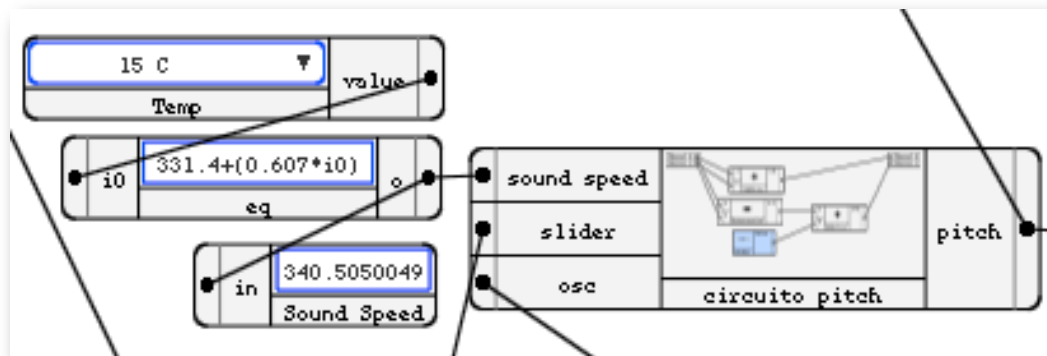
nivel de presión sonora emularía un movimiento de acercamiento y alejamiento de la fuente en estéreo.

A medida que se aleja la fuente, el nivel de presión y el tono que percibe el receptor disminuyen, en la misma proporción en que se percibió al acercare, esto si la velocidad se mantiene constante, de esta manera funcionara el *plugin*. También se relacionó la velocidad de la oscilación en el estero, con el cambio tonal. Si la fuente solo oscila entre los canales del estero, el pitch percibido será el mismo en el centro de la oscilación y disminuirá al alejarse del eje de oscilación.

Si la fuente se desplaza en los dos ejes vertical (nivel) y horizontal (paneo), el cambio total de pitch estaría definido como una suma entre la velocidad del oscilador 1 y la del oscilador 2. Se habilitó la opción de desactivar el cambio de pitch producido por la oscilación en estéreo.

Por último el cambio de tono estará también determinado por la temperatura, entre menor la temperatura mayor ser el cambio percibido, siguiendo la teoría del físico australiano, Cristian Doppler, y es menor si la temperatura aumenta, ya que la velocidad del sonido también aumenta. A continuación se muestra el circuito utilizado para la relación en que la temperatura afecta el cambio de pitch.

Figura 16. Diagrama para factor de cambio de temperatura



DC Doppler, SonicBirth

5.8.3.1 Calibración del Cambio Doppler para la velocidad de los osciladores. Para poder relacionar la velocidad de los osciladores con una velocidad en kilómetros por hora, fue necesario calibrar esta relación de tal manera de que fuera realista el cambio percibido.

En primer lugar se realizaron cálculos para determinar cuál sería el cambio en cents necesario en el oscilador de nivel para emular el efecto Doppler de maneras realistas relacionando las velocidades. Sabiendo que este oscilador tiene una frecuencia de 1 Hz, se puede determinar que el periodo mínimo que tendría la fuente para completar una trayectoria sería de 1 segundo. La

trayectoria completa para este oscilador sería desde que el receptor no percibe la fuente y esta se empieza a acercar, cuando la fuente está justo enfrente de él, hasta que se aleja por completo y ya no es percibida por el receptor. En el eje del panning la trayectoria máxima que recorrería la fuente sería lo que demora la fuente en pasar del canal izquierdo al derecho, aunque esta puede ser determinada por el control de amplitud del oscilador de panning.

Para determinar una distancia en el ambiente de simulación para el efecto Doppler, se calibró un nivel para una fuente que tiene 120 dB a 1 metro del observador con 0 dBfs en el editor de audio, es decir este sería justo enfrente en el espacio acústico. Utilizando la ley del inverso cuadrado, se puede saber que la fuente puntual disminuiría 6 dBs al duplicar la distancia es decir 6 dB cada metro para nuestra calibración, lo que completaría un total de 42 metros para que la fuente llegara a un nivel de 0 dBs en el espacio acústico, o -∞ dBfs.

De esta manera se llega a la conclusión que la fuente emularía una velocidad de 41 m/s en realizar una trayectoria completa en el eje Y (nivel). Así basándose en las fórmulas para efecto Doppler se determinó que en esta velocidad se obtendría a un cambio Doppler de **222.12 cents**, el intervalo que produciría un desplazamiento de fuente a una velocidad de **41m/s** esto en una sola dirección, es decir al acercarse o alejarse únicamente.

De manera similar se determinó el cambio que producirá un movimiento entre los canales izquierdo y derecho del estero, se calibró 120 dB a 1 metro con la posición central y utilizando en la escala anterior se determinó que el espacio entre la posición central y los extremos era de 20.5 m, la velocidad máxima de oscilación de extremo a extremo es de 0.5 Hz, lo que daría que para el espacio virtual del *plug-in* la fuente tendría un cambio en de pitch máximo de **111.06 cents**, poco más de un semitono.

El valor del intervalo total del cambio Doppler en un movimiento combinado en los dos ejes, estará determinado las velocidades ambos osciladores, siguiendo la teoría original del efecto Doppler. Este intervalo será multiplicado por una función que resulta de la suma de la función que describe el cambio de pitch producido por el movimiento en el eje vertical y la que describe el cambio producido en el eje horizontal.

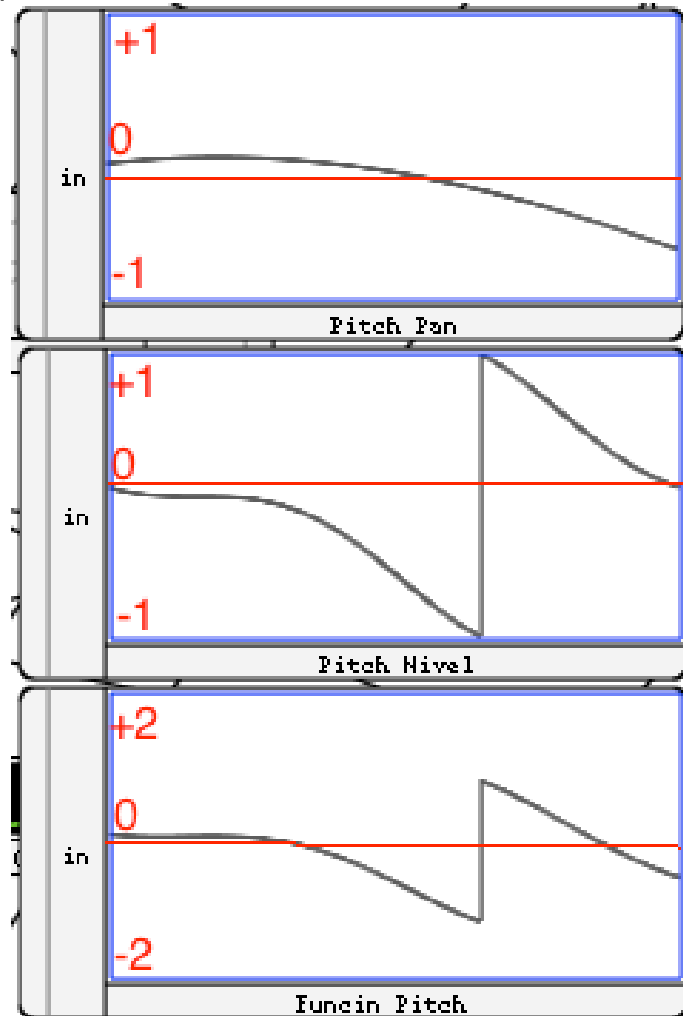
Ecuación 17. Intervalo de pitch total en el AU

$$IDC = ((V_{osc1} * 222,12) + (V_{osc2} * 222,12)) * (FactorT)$$

Donde IDC sería el intervalo Doppler en cents, Vosc1 como la velocidad del oscilador uno en Hz, Vosc2 como la velocidad del oscilador dos en Hz, y el Factor de T, como el factor de la temperatura.

La función que determina el cambio de pitch sería la suma entre la función de cambio de pitch producida por el paneo, y la función de cambio de pitch producida por el cambio de nivel, como se aprecia en la figura 17.

Figura 17 Ejemplo función de cambio Pitch



Desarrollo de *AudioUnit* de efecto Doppler, SonicBirth.

5.8.3.2 Sincronización de osciladores de Pitch y Trayectoria Para sincronizar los osciladores de cambio de pitch, paneo y nivel fue necesario manipular sus fases, además de esta manera se crearon señales complejas a partir de las señales sencillas de los generadores, para emular correctamente el cambio de pitch que describirían los diferentes movimientos permitidos para la oscilación de la fuente.

La frecuencia de los osciladores es la misma, esta determina la velocidad de la fuente, pero las fases varían, para el generador de senoidales se utilizó una fase de $\pi/2$ en radianes, o 90° y para los de triangulares y pitch, se utilizó la misma fase de π o 180° .

Figura 18. Ejemplo de osciladores de nivel y cambio de pitch por OSC 1, en sincronía

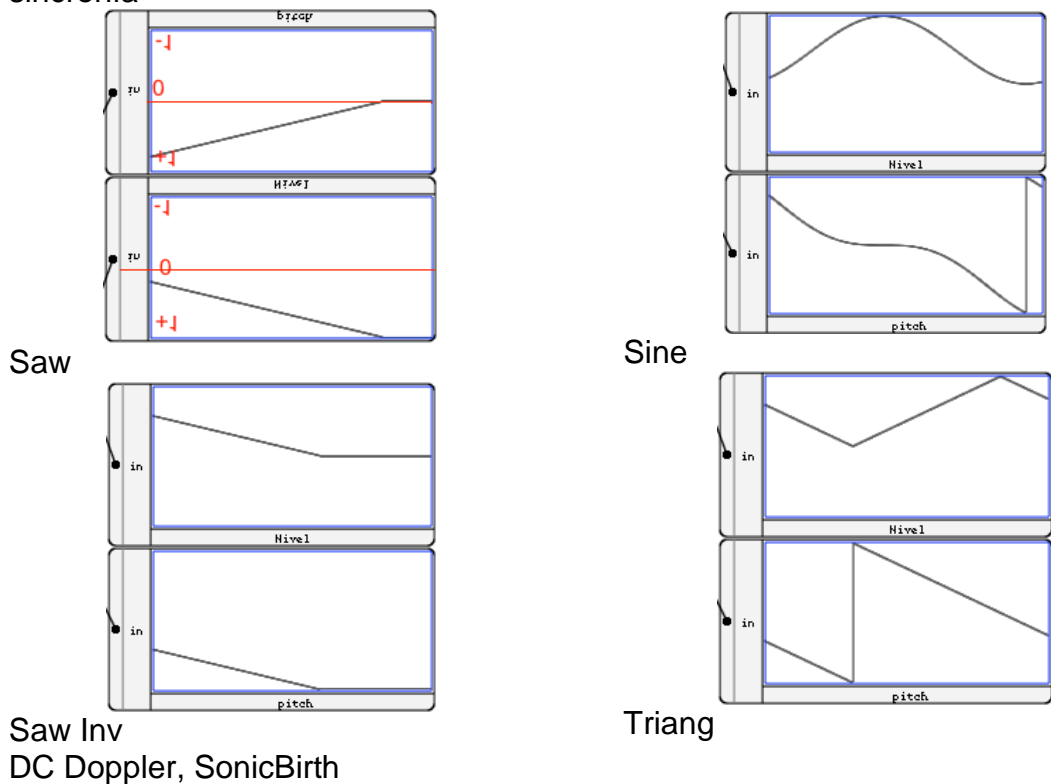
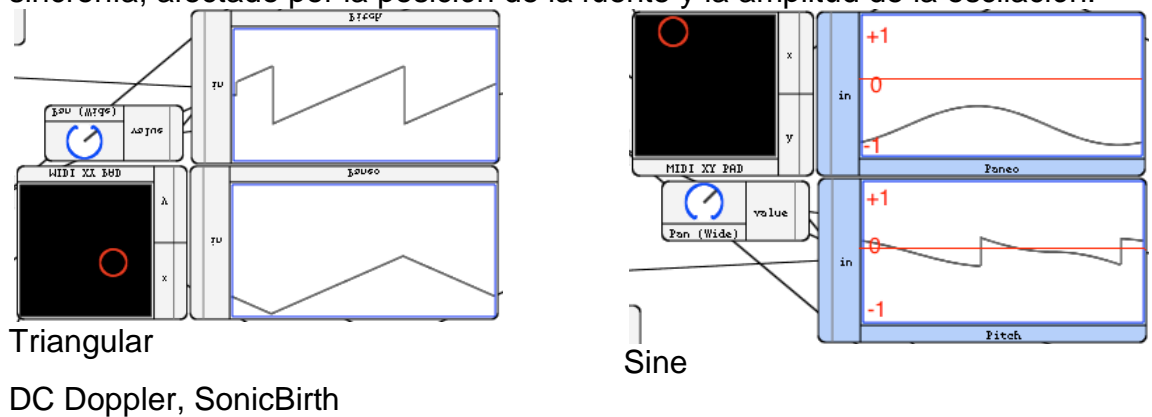


Figura 19. Ejemplo osciladores de paneo y cambio de pitch por OSC 2 en sincronía, afectado por la posición de la fuente y la amplitud de la oscilación.



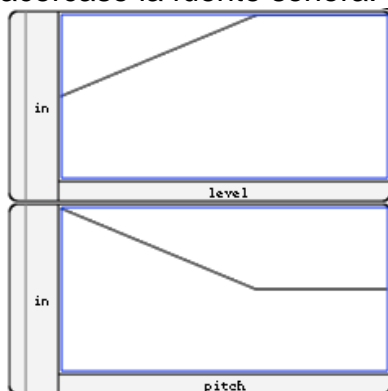
En los anexos 1 y 2 se muestran los diagramas para las opciones del oscilador de paneo, pitch y nivel.

5.8.3.3 Movimiento tipo "One Shot" Para describir un movimiento que no fuera necesariamente oscilatorio se agregó la opción de movimiento "One Shot". Este movimiento se activa cuando el usuario selecciona la opción de forma de onda tipo Saw o Sierra, para el oscilador de nivel.

Cuando esta opción es seleccionada el movimiento solo será de acercamiento, o alejamiento, al acercarse la fuente vendrá disminuyendo su pitch desde la frecuencia Doppler, determinada por la velocidad, hasta su frecuencia original. Esto se logró mediante un "trigger" que dispara un valor de 1, cuando la señal alcanza un valor de 1. La señal original de los osciladores para este movimiento será una diente de cierra multiplicada por 0.5 y sumada por 0.5 para que alcance los valores de 0 a 1 únicamente.

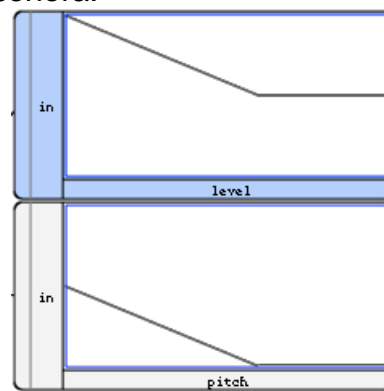
Estas funciones son manipuladas de tal manera que al multiplicar sus valores por el pitch y nivel de la señal original, permita emular el movimiento de acercamiento o alejamiento de la fuente aplicando un cambio Doppler de pitch. También se habilitó esta opción para el *paner*. En las imágenes 20 y 21 mostradas a continuación, se puede observar la manera en que trabajan los osciladores de pitch y nivel en el moviendo ".One Shot".

Figura 20. Comportamiento de los osciladores de pitch y nivel al acercarse la fuente sonora.



DC Doppler, SonicBirth

Figura 21. Comportamiento de los osciladores de pitch y nivel al alejarse la fuente sonora.



DC Doppler, SonicBirth

5.8.3.4 Opciones Doppler Start / Stop, Trigger / Reset y Sync Oscilators.

Finalmente para el circuito del DSP, se desarrolló una opción de Doppler *Start / Stop*, y *Trigger / Reset* que permite al usuario disparar el movimiento de la fuente, detenerlo en cualquier momento y hacer que el desplazamiento vuelva a iniciar. Esto se hizo utilizando un elemento de *Trigger* dentro de la herramienta SonicBirth que, envíe un valor de frecuencia cero a los generador lo que detiene el movimiento, o envía el valor de la frecuencia determinada por el usuario para que inicie su movimiento.

Se utilizó una variable "booleana", es decir que solo emite valores de verdadero o falso, para la opción de Doppler *Start / Stop*, multiplicando esta valor por el de la frecuencia de los osciladores. Al ser cero la frecuencia en todos los osciladores será de cero y el oscilador se detendrá, y al ser uno la frecuencia en los osciladores volverá a ser la deseada por el usuario y el oscilador volverá a entrar en actividad.

La opción de *trigger / reset*, fue desarrollada para controlar el movimiento "One Shot" descrito anteriormente, permite que en cualquier momento el usuario active el *trigger* que pondrá en movimiento la fuente, además este mismo control permite que el usuario haga que la fuente vuelva a su posición original, y así sucesivamente, si es deseado. Esta función fue desarrollada utilizando, un control de SonicBirth llamado "*keyboard tap*", que emite un valor de "1" en su salida, al ser activado, no puede ser controlado por MIDI. Para esto también se utilizaron elementos comparadores y el elemento *trigger* en SonicBirth. Al activar el *trigger* este emite un uno, que es inmediatamente duplicado y reinicia los *triggers* de los osciladores de nivel y pitch y paneo, para el movimiento "One Shot".

También se desarrolló la opción de sincronizar los osciladores para que así se pueda obtener un movimiento más controlado del movimiento entre izquierda y derecha, nivel y pitch. Al habilitar esta opción se controla la velocidad de los osciladores únicamente con el control de la frecuencia del oscilador de nivel. Se agregó un menú para elegir diferentes relaciones entre las velocidades de los osciladores, en relación desde 1 a 1 veces, hasta 4 a 1, la velocidad de la oscilación en el eje vertical sería 4 veces la velocidad en el horizontal.

El circuito final para el DSP, con sus circuitos y sub-circuito es mostrado en el **Anexo 3**.

5.9 DESARROLLO DE LA INTERFAZ GRÁFICA DE USUARIO

SonicBirth es una herramienta que permite el desarrollo de una interfaz de usuario personalizada de manera muy sencilla intuitiva, ideal para un ingeniero de sonido.

Al elegir la opción de "*Custom GUI*" (interfaz gráfica de usuario personalizada) en la configuración del circuito, se habilita la opción para desarrollar la interfaz de usuario personalizada. Esta función le permite elegir el tamaño deseado para el *plug-in* en "Píxeles", en este caso el *plug-in* tendrá una dimensión de 600X450 píxeles,

También se utilizaron colores personalizados para los controles de los parámetros del *plug-in* en la interfaz gráfica de usuario, y una imagen de fondo personalizada. Además permite la opción de agregar imágenes personalizadas para los controles como *knobs* y *faders* y el *Pad XY*.

Para ubicar los controles en la GUI, se habilita la opción "*GUI Design*", que se ubica en la parte inferior del menú de configuración del circuito, de esta manera SonicBirth habilita al programador para organizar todos los elementos utilizados en el circuito.

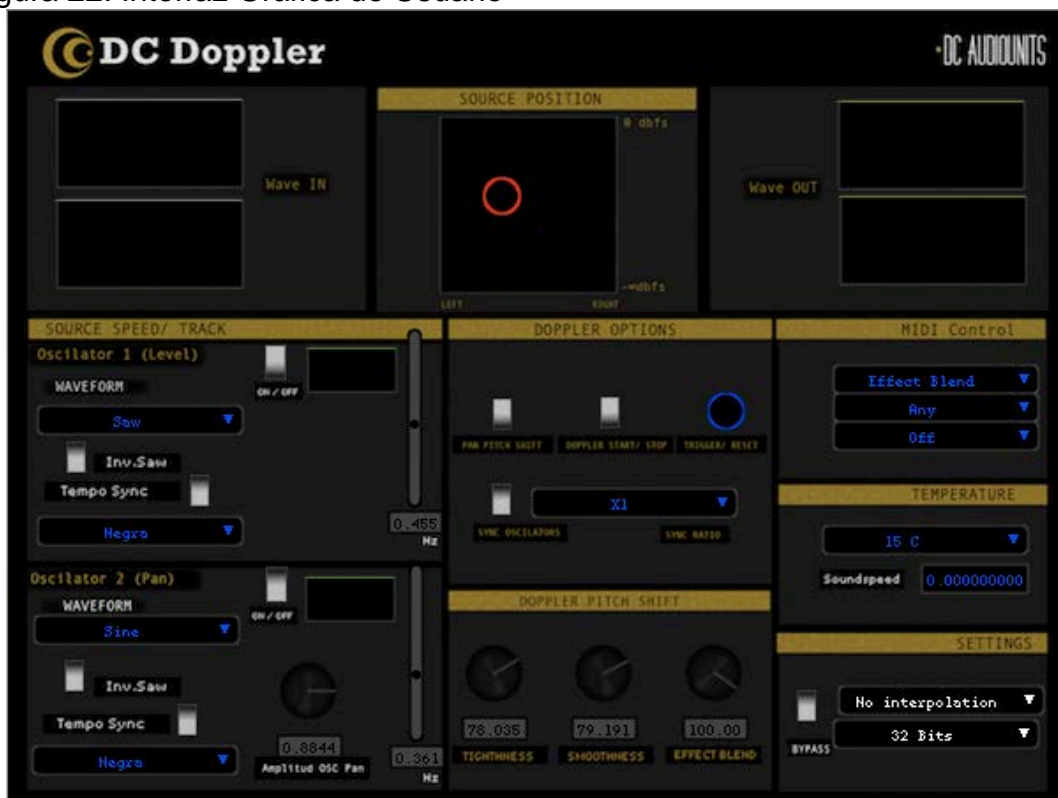
Para este circuito del DC Doppler, nombre que se utilizó para el *plug-in* en el editor e interfaz de usuario, se utilizaron elementos gráficos en la interfaz de usuario para proveer una información más visual acerca del comportamiento de

los osciladores de "Pitch, Nivel y Paneo." Se ubicaron osciloscopios en la entradas y salidas estereofónicas del *plug-in*, para permitir que el usuario vea el cambio en la forma de onda de la señal que entra es afectada por el procesamiento.

También se utilizaron do osciloscopios a la salida de los osciladores de nivel y paneo, para mostrar la forma de onda que haya sido elegida por el usuario, y de qué manera estos osciladores están funcionando.

A continuación se muestra la interfaz de usuario personalizada.

Figura 22. Interfaz Gráfica de Usuario



DC Doppler, DC Audio Units

6. PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

El *audio unit* que emula el efecto Doppler será analizado de cuatro maneras diferentes, la primera será obviamente, el análisis del funcionamiento dentro del *software* de desarrollo **SonicBirth**, en segundo lugar se realizara un análisis del funcionamiento en **AU Lab** de Apple esta herramienta es de uso gratuito y hace parte del kit de desarrollo XCodeTools, de esta misma compañía, un *software* para realizar pruebas de funcionamiento de *audio units*. Una vez superadas estas dos fases, se procederá a utilizar el Audio Unit en un DAW común que utilice *audio units*, en esta caso **Garage Band** y **Audacity**, ya que es de uso gratuito para usuarios de computadores Apple, esto para experimentar el funcionamiento normal que tendría un usuario, con automatización y control MIDI.

Finalmente se realizarán mediciones del cambio Doppler Producido por el *plug-in* utilizando una herramienta desarrollada con este propósito, para correlacionar los datos.

6.1 ANÁLISIS DEL FUNCIONAMIENTO DEL DOPPLER EFFECT, EN AULAB Y SONICBIRTH

Tanto AULab como SonicBirth permiten comprobar el funcionamiento del *plugin* fuera de un editor, de esta manera se así analizará y comprobará el funcionamiento de todos sus parámetros, control MIDI, carga en la CPU etcétera. En estos *software* se realizarán las primeras pruebas de funcionamiento del audio unit de efecto Doppler.

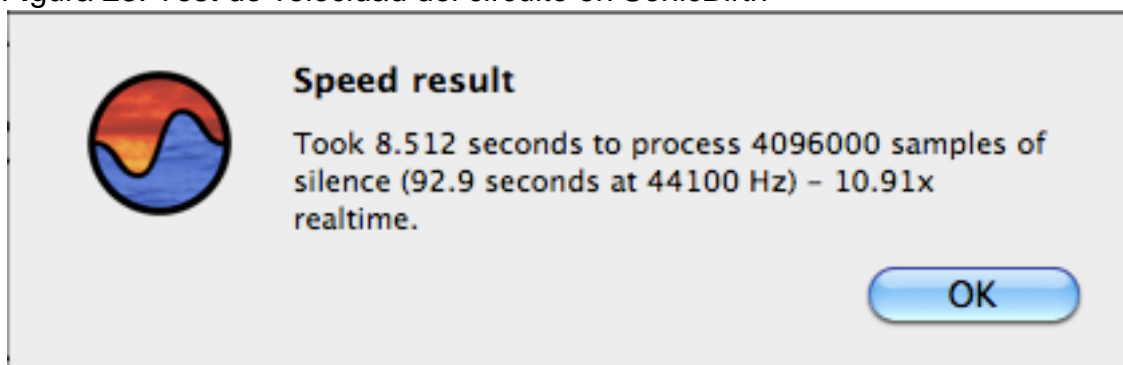
6.1.1 Velocidad de procesamiento del Audio Unit en SonicBirth Las pruebas iniciales de todos los componentes, circuito y sub-circuitos se realizan utilizando SonicBirth a medida que se desarrolla el *plug-in*, se comprueba el funcionamiento que se experimentará al utilizar el Audio Unit en AU Lab o en un DAW al manipular la interfaz de usuario.

SonicBirth permite medir la velocidad total en que trabaja el circuito, midiendo el tiempo que demora en procesar un número determinado de muestras. A una frecuencia de muestreo de 44100 Hz También se puede saber su carga en la CPU. Este es el primer análisis del *plug-in*.

Se reproduce un archivo de audio en formato **wav**, se comprueba el funcionamiento final, manipulando los elementos como un usuario lo haría al utilizar el *plug-in* en un DAW. La configuración final dentro de SonicBirth al compilar el audio unit será la que queda establecida como la original, por "default". También se puede exportar en formato VST.

En la imagen 23 se muestra el test de velocidad obtenido para el circuito final del Audio Unit.

Figura 23. Test de velocidad del circuito en SonicBirth



SonicBirth

Carga de la CPU en SonicBirth:

```
Audio cpu usage: 16.1 % , Device: 2 channels 48000 hz 100 millisecs .
```

6.1.2 Funcionamiento y control del Audio Unit en AU Lab Una vez el Audio Unit es reconocido por la terminal, el Audio Unit está listo para ser ejecutado en AU Lab o cualquier DAW que trabaje con *audio units*. AU Lab es una herramienta que permite ejecutar los *audio units* afuera de un DAW para realizar pruebas de funcionamiento, es gratuito y viene incluido en dentro de XCode Tools de Apple.

Primero es necesario establecer la señal de entrada al *audio unit*. Se utilizó en este caso un *audio unit* Reproductor de Audio, incluido a AU Lab. Para las primeras pruebas se utilizó un archivo de audio formato .wav, a 44.1kHz 16 Bit.

La capacidad del computador en que se realizaron los análisis es de 2.16 gigahertz de procesador, con doble núcleo y 3 gigabytes de RAM.

Se comprobó el correcto funcionamiento de los osciladores de paneo, ganancia y pitch, aunque fue necesario realizar bastantes correcciones y optimización del circuito una vez fue probado en AU Lab para tratar de llegar a un control óptimo del *audio unit* sin necesidad de detener el motor de audio, aunque no fue posible hacerlo para todos los parámetros.

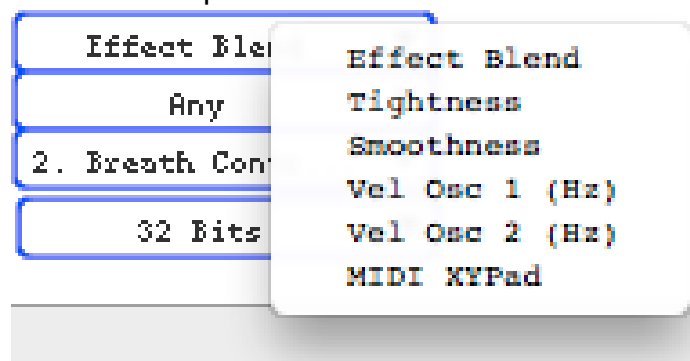
Además se encontró que es necesario detener el motor de audio para cambiar algunas de las opciones del *audio unit* ya que no es posible hacerlo en tiempo real, como el cambio en la forma de onda, la opción de sincronizar los osciladores y el cambio entre el control por osciladores o por *Pad* MIDI XY.

6.1.3 Control de parámetros por MIDI en AU Lab y SonicBirth Inicialmente se probaron las funciones básicas de control MIDI desarrolladas para el Audio Unit utilizando AU Lab, obteniendo un comportamiento adecuado en el control

de los parámetros desarrollados para ser controlados por MIDI, aunque el DAW Garage Band de Apple, permite asignar un control MIDI a prácticamente todos los parámetros disponibles en el *Audio Unit*.

La configuración MIDI del *Audio Unit* de Doppler permite manipular tres opciones, en primer lugar tiene un menú para seleccionar uno de los parámetros diseñados para ser manipulados. Se puede el canal y control MIDI con el cual el parámetro será controlado. Los controles MIDI desarrollados en SonicBirth permiten ser asignados mediante una opción llamada "Learn", que al ser elegida asignará el parámetro seleccionado en el *Audio Unit* al primer control MIDI que sea utilizado.

Figura 24. Controles MIDI disponibles en el Audio Unit



DC Doppler, DC AudioUnits

6.2 ANÁLISIS DEL FUNCIONAMIENTO DEL AUDIO UNIT EN GARAGE BAND Y AUDACITY.

Para que el plug-in sea reconocido debe estar correctamente instalado, para esto se exporta el archivo *.component* a la carpeta de *plug-ins* del usuario dentro de la librería. Para el *Doppler Effect* esta carpeta se encuentra ubicada de la siguiente manera

```
:MacintoshHD/Library /Audio/Plug-Ins/Components/DCDoppler.component
```

En segunda instancia se debe comprobar que el sistema reconozca el *plug-in*, y que haya sido correctamente instalado, para esto es necesario ir a la terminal del sistema e ingresar el código *auval -a*. Al ingresar este comando la terminal permite observar todos los *audio units* instalados en el disco duro, además de esto muestra tres columnas con información acerca del *audio unit*

En la primera columna se observa el tipo de *Audio Unit* desarrollado, en este caso *aufx*, el sub-tipo en la segunda columna, *xfua*, y el código del programador en la tercera, *DCau*.

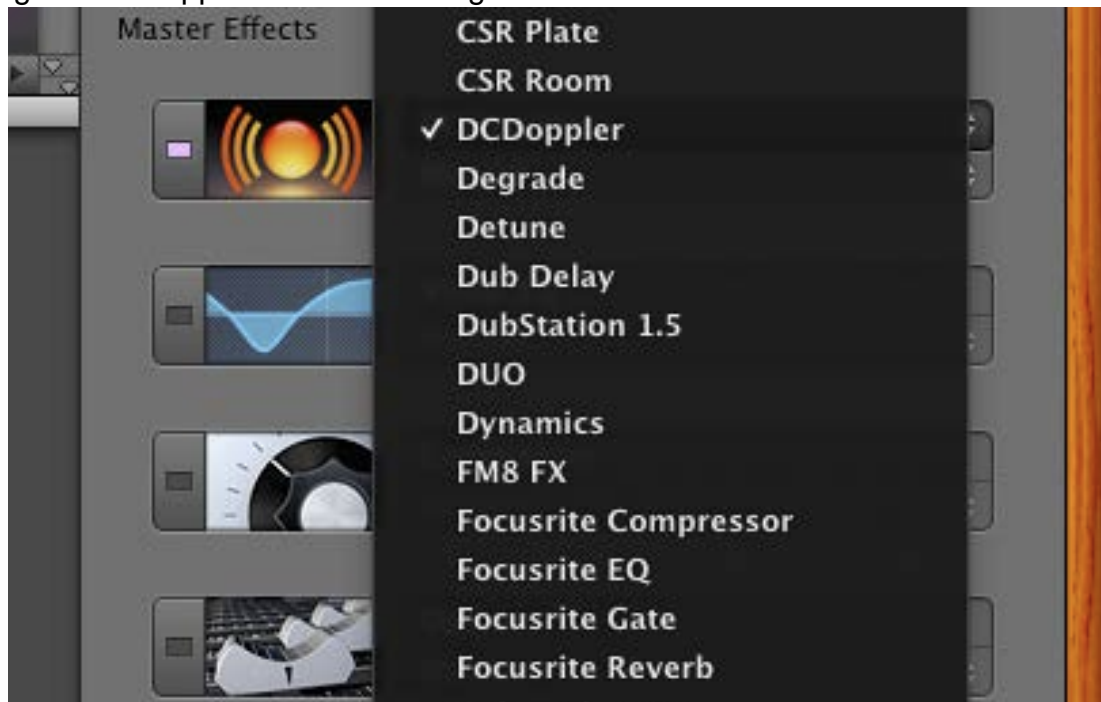
Lista de *plug-ins* mostrados por la terminal:

```
aufx scmp FCUS - Focusrite: Focusrite Compressor
aufx sdly appl - Apple: AUSampleDelay
aufx sfeq FCUS - Focusrite: Focusrite EQ
aufx svrb FCUS - Focusrite: Focusrite Reverb
aufx tmlo DCau - DCaudiounits: Tremolo Unit
aufx tmpt appl - Apple: AUPitch
aufx xfua ScBh - DCau: DCDoppler
aumf ATxE VST - Antares: Auto-Tune Evo
aumf B42S -NI- - Native Instruments: B4 II FX
aumf CWam -NI- - Native Instruments: Absynth 4 FX
```

Como se puede apreciar el *Audio Unit* es reconocido por la terminal, lo que confirma la correcta instalación de este en el sistema.

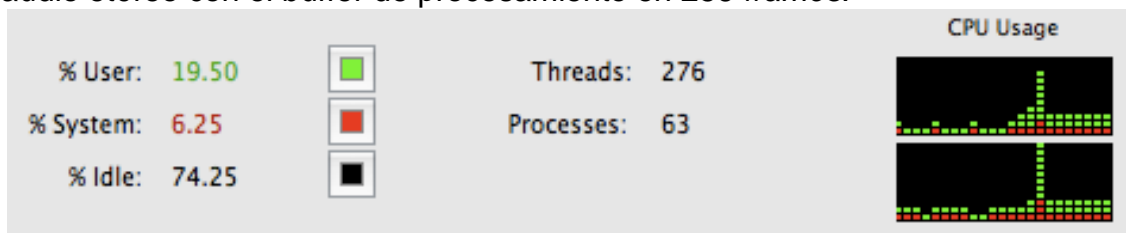
Para comprobar el correcto funcionamiento dentro de una DAW, además de las opciones de automatización y control MIDI, se prueba el *Audio Unit* en Garage Band, en este *Host* es posible utilizar todos los parámetros establecidos anterior mente por medio de MIDI y es de acceso gratuito para usuarios de computadores Apple.

Figura 25. Doppler Effect en Garage Band

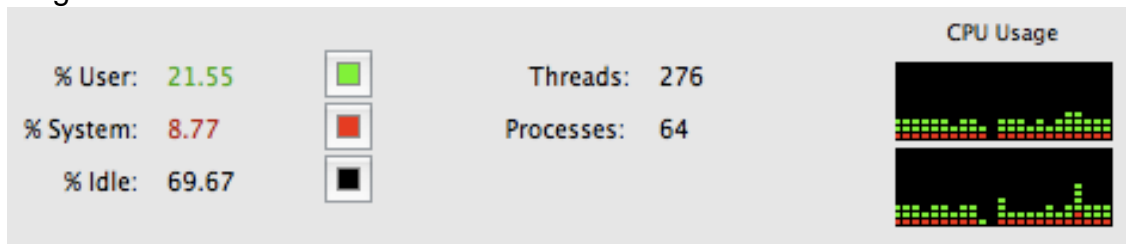


Garage Band

Figura 26 Carga de la CPU en el monitor de actividad al utilizar el *plug-in* en Audacity a diferentes frecuencias de muestreo, procesando un solo canal de audio etéreo con el buffer de procesamiento en 256 *frames*.



Carga a 44.1000 Hz de frecuencia de muestreo.



Carga a 48.0000 Hz de frecuencia de muestreo.

El comportamiento del plugin en el DAW es adecuado, produce una carga aceptable en el sistema a una resolución hasta de 48KHz. El tiempo de procesamiento del software es adecuado, aunque es necesario de una CPU de gran capacidad para trabajar el *plug-in* en frecuencias de muestreo superiores a 48000 Hertz. Se puede ver que en frecuencias de muestreo superiores a esta se presenta una carga excesiva la CPU.

6.2.1 Automatización de Parámetros y Control MIDI Se comprobó la opción de automatización para todos los parámetros y opciones que permite el Audio Unit. En el DAW es posible detener el motor de audio, permitiendo a los osciladores se sincronizarse al cambiar los parámetros del *plug-in*, y de esta manera se consigue un funcionamiento adecuado.

Se analizaron las funciones de automatización para todos los parámetros obteniendo un resultado satisfactorio.

Todos los parámetros son posibles de automatizar, aunque es necesario recordar que no todas las automatizaciones son posibles, como cambiar la forma de onda de los osciladores o sincronizarlos en tiempo real, ya que es necesario detener el motor de audio al cambiar estos parámetros, para que los osciladores trabajen como el usuario desee.

A diferencia de AULab y SonicBirth, el control por MIDI del Audio Unit de Doppler en Garage Band es total, es decir todos los controles pueden ser asignados a controles MIDI a excepción de la opción "Trigger / Reset" que no es posible de asignar ni autorizar en ningún software. Esto se debe a la

configuración inicial del elemento "Keyboard Tap" de Sonic Birth utilizada en el *audio unit* para esta función.

6.3 DESARROLLO DEL INSTRUMENTO MEDICIÓN PARA LA APLICACIÓN DESARROLLADA, CORRELACIONANDO LOS DATOS CON MEDICIONES OBJETIVAS.

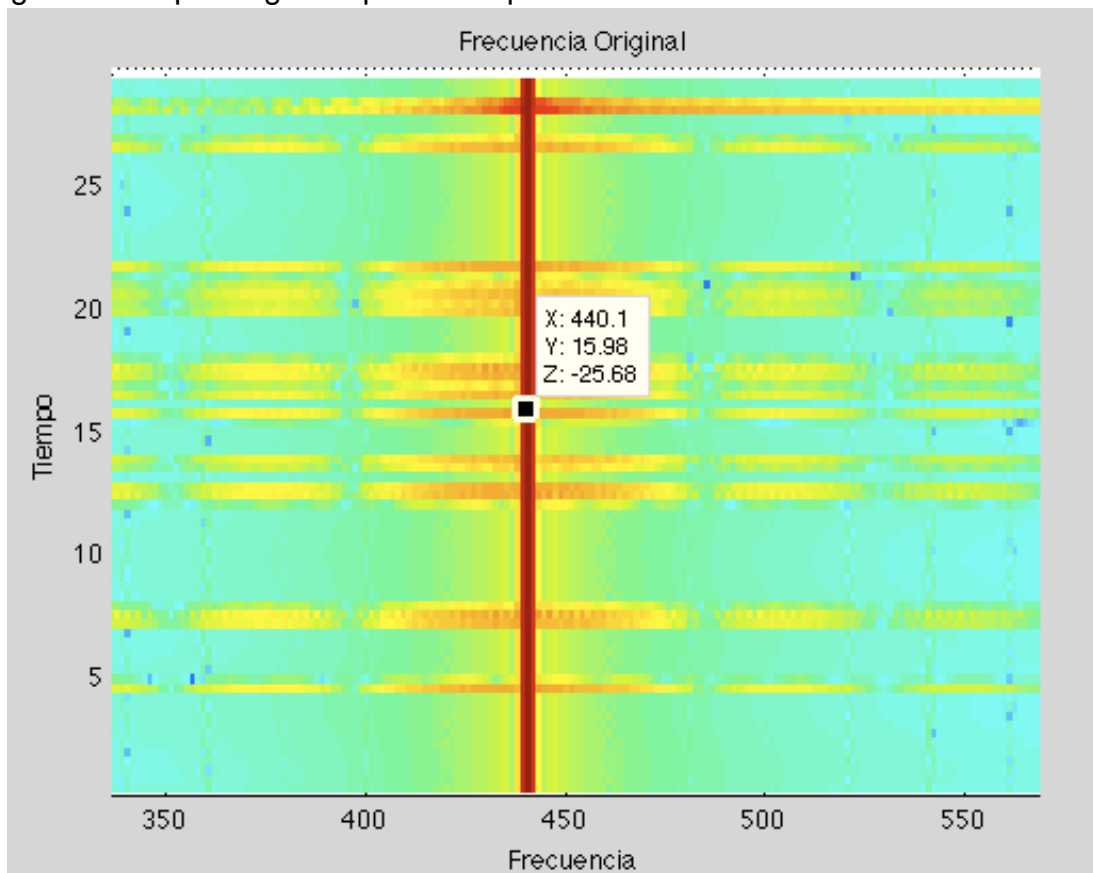
6.3.1 Desarrollo de una herramienta de medición para el *plug-in* DC Doppler AU. Para medir el cambio de tono obtenido al utilizar el *plug-in* de Efecto Doppler a diferentes velocidades en los osciladores de nivel y ganancia, y en las diferentes trayectorias que permite el *plug-in*, se desarrolló un programa en MatLab, que permite observar el contenido espectral de una señal por medio de la implementación de un espectrograma. En este se puede observar la señal en frecuencia, tiempo y amplitud.

Para tener una medición precisa del intervalo Doppler es necesario el uso de una señal de un tono puro invariable en el tiempo, aunque también se mide el cambio producido en señales complejas, aunque no es posible calcular un intervalo exacto por que el sonograma muestra demasiada información, y no es posible leer claramente un intervalo exacto. En el anexo 4 se muestra el código desarrollado en MatLab para la herramienta de medición.

Criterios de la medición: Se exportaron archivos monofónicos desde Garage Band, en formato *wav* a 44100 Hz de frecuencia de muestreo y 16 Bits, con diferentes señales, un tono puro de 440 Hz, 1 kHz y una señal compleja, en este caso un saxofón, procesadas por el *AudioUnit*, se hizo de esta manera para evitar que la señal procesada fuera afectada por el medio acústico y tener así una lectura clara de lo que está sucediendo. En los resultados también se muestra la interfaz de usuario para el *plug-in* en cada medición. Se utilizó una señal monofónica para medir en la herramienta. Si se quiere analizar los dos canales es necesario sacar una gráfica para cada una, aunque con el archivo mono se puede observar claramente el cambio de tono.

Todos estos resultados se obtienen para el diseño y calibración de este *plug-in*, en otros desarrollos se podrían obtener resultado diferentes. A continuación se muestra la medición de la frecuencia original sin ser procesada por el *plug-in*, como se puede observar es un tono senoidal puro de 440 Hz.

Figura 27. Espectrograma para tono puro 440 hz.



Herramienta de medición.

En la gráfica anterior se puede observar un línea roja en la frecuencia que se está midiendo, en este caso 440 Hz, en el eje X se puede la escala en frecuencia, en el eje Y se observa el tiempo de duración de la muestra en este caso 30 segundos, se puede observar que la muestra no varía en el tiempo.

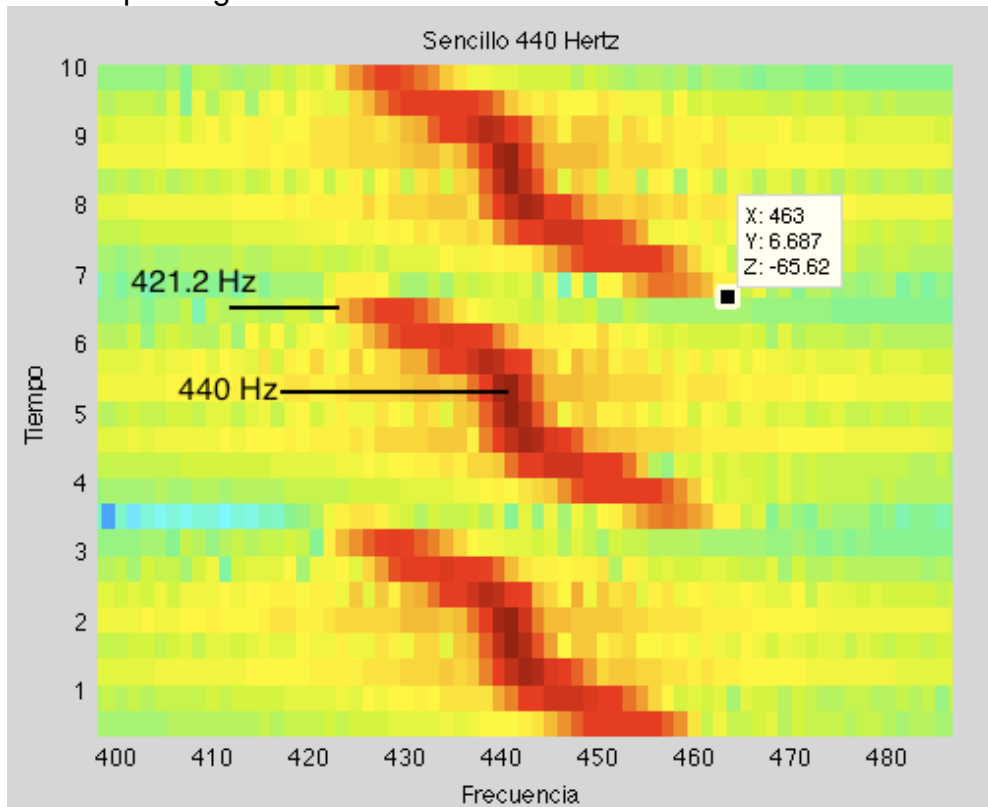
6.3.2 Medición del cambio Doppler para movimiento sencillo en el eje Y. Utilizando una frecuencia fundamental de 440 Hz. El siguiente es un ejemplo de una medición que se realizó para un movimiento en el eje Y, es decir determinado únicamente por el oscilador de nivel a una velocidad de oscilación de 0.4 Hz, la fuente tarda 2.5 segundos en completar una trayectoria en este eje (alejarse por completo y volver a estar justo en frente del observador).

Figura 28. Vista de usuario del *plug-in* para la medición de un movimiento en eje Y en 440 Hz y 1kHz.



Interfaz de usuario.

Figura 29. Espectrograma movimiento sencillo 440 Hz

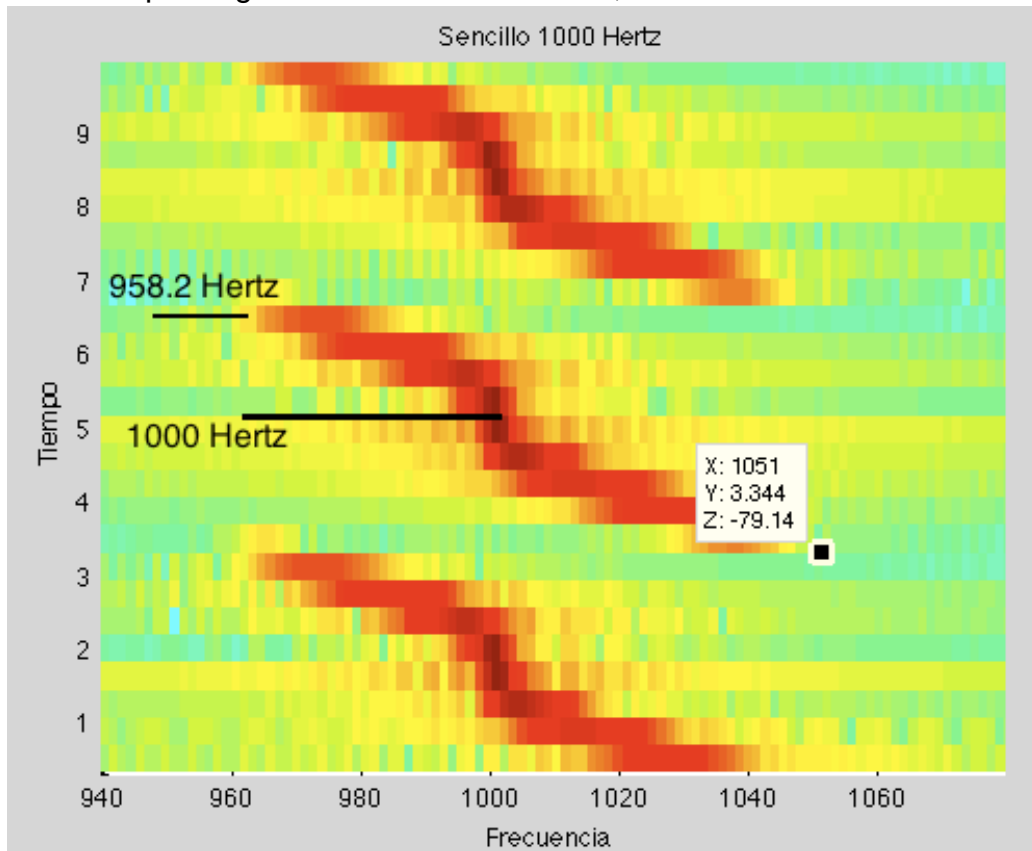


Herramienta de medición

En la imagen 29 se puede observar el cambio Doppler a una frecuencia de oscilación en el eje Y de 0.4 Hz. Se pueden observar la frecuencia fundamental y las dos frecuencias Doppler de 421.1 Hz y 463, el intervalo Doppler en este caso es de alrededor de 88.7 cents. También se puede observar el tiempo en que dura una oscilación. Los colores muestran la amplitud de la señal, los colores más intensos como rojos muestran un nivel mayor, el amarillo y el verde representan niveles más bajos en este orden, y el azul sería silencio. La frecuencia de la izquierda 421.2 representa la frecuencia que tendría al alejarse por completo, la de la derecha y la que tendría cuando la fuente se empieza a acercar, y la central muestra la frecuencia fundamental, esto se puede comprobar con la escala del tiempo.

6.3.3 Medición del cambio Doppler para movimiento sencillo en el eje Y. Utilizando una frecuencia fundamental de 1000 Hz. Para comprobar que el cambio es uniforme en todo el espectro se midió también la frecuencia Doppler en esta misma velocidad para una frecuencia fundamental de 1000 Hz, en este caso el intervalo es de 86.12 cents, se produce un margen de error de diferencia de 1 cent, posiblemente por falta de decimales en las frecuencias mostradas por MatLab. La configuración para el *plug-in* es la misma que para el ejemplo para un movimiento sencillo en 440 Hertz.

Figura 30. Espectrograma movimiento sencillo, 1000 Hertz



Herramienta de medición

6.3.4 Tabla de cambio Doppler para el Oscilador de Nivel En base a estas mediciones se realizó la siguiente tabla.

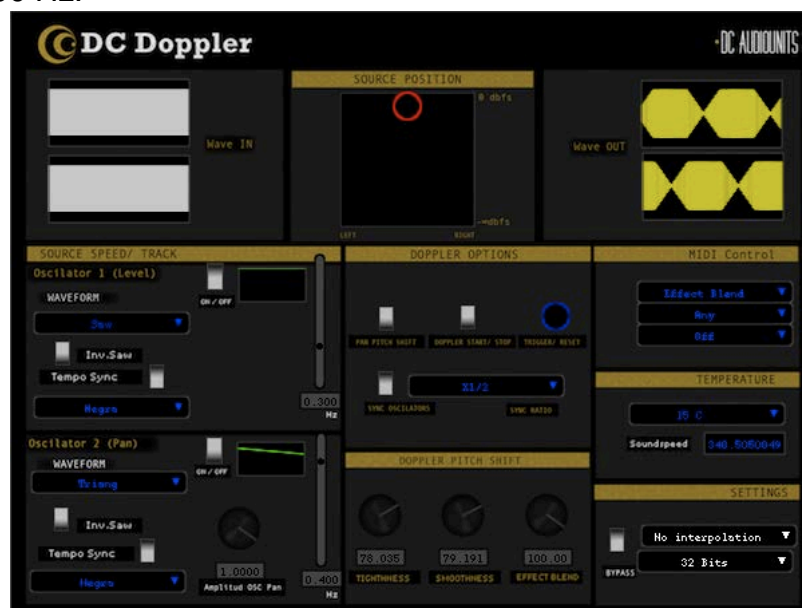
Tabla 10. Intervalo del cambio Doppler en Cents obtenidos al utilizar el plug-in, en un movimiento sencillo en el eje Y, a diferentes velocidades y frecuencias.

Velocidad Osc 1	64 Hz	125 Hz	250 Hz	500 Hz	1,000 Hz
0.01 Hz	2.22 Cents	2.22 Cents	2.22 Cents	2.22 Cents	2.22 Cents
0.02 Hz	4.44 Cents	4.44 Cents	4.44 Cents	4.44 Cents	4.44 Cents
0.03 Hz	6.66 Cents	6.66 Cents	6.66 Cents	6.66 Cents	6.66 Cents
0.04 Hz	8.88 Cents	8.88 Cents	8.88 Cents	8.88 Cents	8.88 Cents
0.05 Hz	11.1 Cents	11.1 Cents	11.1 Cents	11.1 Cents	11.1 Cents
0.06 Hz	13.27 Cents	13.27 Cents	13.27 Cents	13.27 Cents	13.27 Cents
0.07 Hz	15.54 Cents	15.54 Cents	15.54 Cents	15.54 Cents	15.54 Cents
0.08 Hz	17.76 Cents	17.76 Cents	17.76 Cents	17.76 Cents	17.76 Cents
0.09 Hz	19.99 Cents	19.99 Cents	19.99 Cents	19.99 Cents	19.99 Cents
0.1 Hz	22.21 Cents	22.21 Cents	22.21 Cents	22.21 Cents	22.21 Cents
0.2 Hz	44.42 Cents	44.42 Cents	44.42 Cents	44.42 Cents	44.42 Cents
0.3 Hz	66.63 Cents	66.63 Cents	66.63 Cents	66.63 Cents	66.63 Cents
0.4 Hz	88.87 Cents	88.87 Cents	88.87 Cents	88.87 Cents	88.87 Cents
0.5 Hz	111.06 Cents	111.06 Cents	111.06 Cents	111.06 Cents	111.06 Cents
0.6 Hz	133.27 Cents	133.27 Cents	133.27 Cents	133.27 Cents	133.27 Cents
0.7 Hz	155.48 Cents	155.48 Cents	155.48 Cents	155.48 Cents	155.48 Cents
0.8 Hz	177.69 Cents	177.69 Cents	177.69 Cents	177.69 Cents	177.69 Cents
0.9 Hz	199.9 Cents	199.9 Cents	199.9 Cents	199.9 Cents	199.9 Cents
1 Hz	222.1 Cents	222.1 Cents	222.1 Cents	222.1 Cents	222.1 Cents

Medición de *plug-in*.

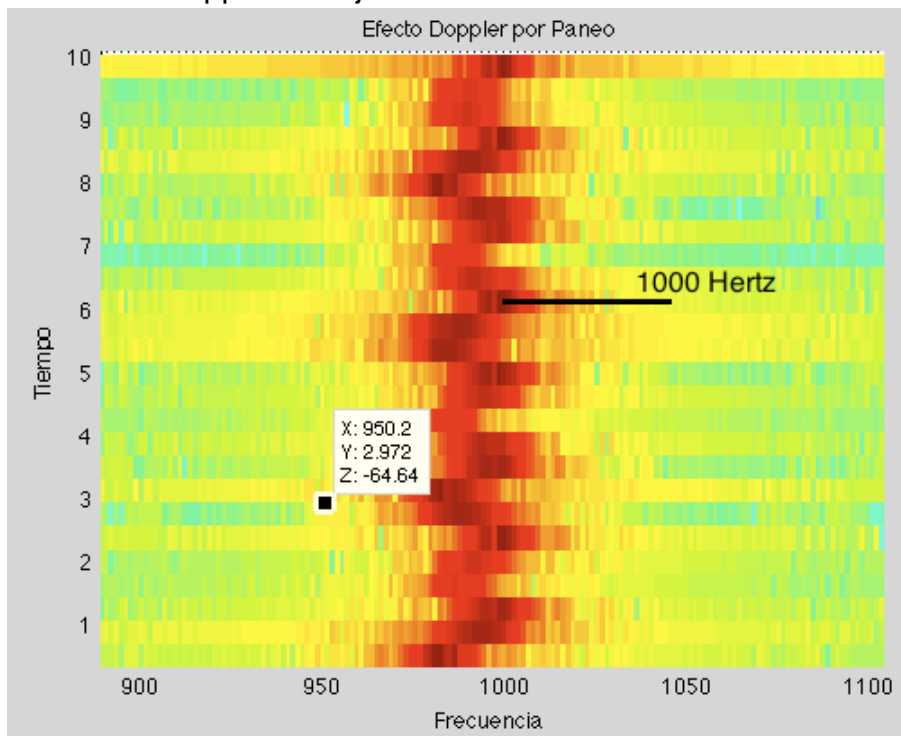
6.3.5 Medición del cambio Doppler para movimiento sencillo en el eje X. Con una frecuencia fundamental de 1000 Hz. En el moviendo con el oscilador de paneo, la fuente se mide de la misma manera, pero esta vez la frecuencia original se percibe cuando la fuente esta justo en el centro del movimiento del paneo. La relación en que el paneo afecta el pitch es menor, por lo que el intervalo a diferentes velocidades será , debido a la calibración que se le dió al *plug-in*.

Figura 31. Vista de usuario del *plug-in* para la medición de movimiento en el eje X, para 1000 Hz.



Interfaz de usuario.

Figura 32. Cambio Doppler en eje X.



Herramienta de Medición.

En la imagen anterior se puede ver el cambio Doppler producido por una oscilación en el eje X a una velocidad de 0.4 Hertz en una la dirección descendiente; el intervalo Doppler medido fue de 88.44 cents.

6.3.6 Tabla de cambio Doppler producido por el *plug-in* con el oscilador de Paneo.

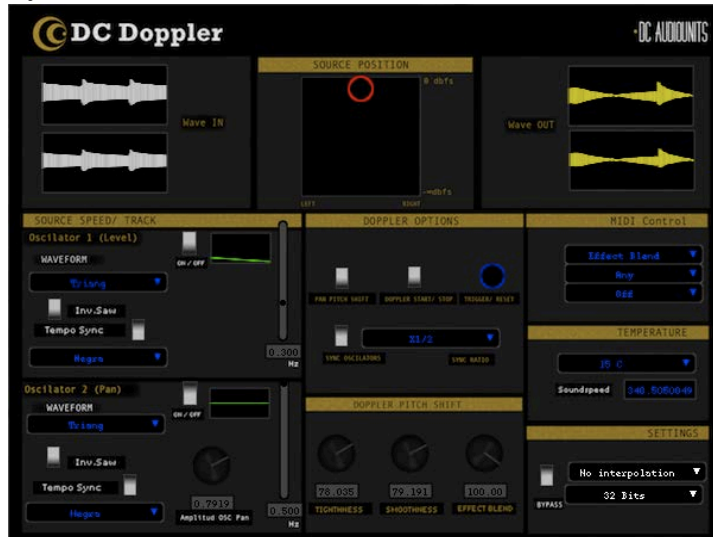
Tabla 11. Intervalo Doppler en el *plug-in* a diferentes velocidades del oscilador 2.

Velocidad Osc 2	64 Hz	125 Hz	250 Hz	500 Hz	1,000 Hz
0.01 Hz	2.22 Cents	2.22 Cents	2.22 Cents	2.22 Cents	2.22 Cents
0.02 Hz	4.44 Cents	4.44 Cents	4.44 Cents	4.44 Cents	4.44 Cents
0.03 Hz	6.66 Cents	6.66 Cents	6.66 Cents	6.66 Cents	6.66 Cents
0.04 Hz	8.88 Cents	8.88 Cents	8.88 Cents	8.88 Cents	8.88 Cents
0.05 Hz	11.1 Cents	11.1 Cents	11.1 Cents	11.1 Cents	11.1 Cents
0.06 Hz	13.27 Cents	13.27 Cents	13.27 Cents	13.27 Cents	13.27 Cents
0.07 Hz	15.54 Cents	15.54 Cents	15.54 Cents	15.54 Cents	15.54 Cents
0.08 Hz	17.76 Cents	17.76 Cents	17.76 Cents	17.76 Cents	17.76 Cents
0.09 Hz	19.99 Cents	19.99 Cents	19.99 Cents	19.99 Cents	19.99 Cents
0.1 Hz	22.21 Cents	22.21 Cents	22.21 Cents	22.21 Cents	22.21 Cents
0.2 Hz	44.42 Cents	44.42 Cents	44.42 Cents	44.42 Cents	44.42 Cents
0.3 Hz	66.63 Cents	66.63 Cents	66.63 Cents	66.63 Cents	66.63 Cents
0.4 Hz	88.87 Cents	88.87 Cents	88.87 Cents	88.87 Cents	88.87 Cents
0.5 Hz	111.06 Cents	111.06 Cents	111.06 Cents	111.06 Cents	111.06 Cents

Resultados

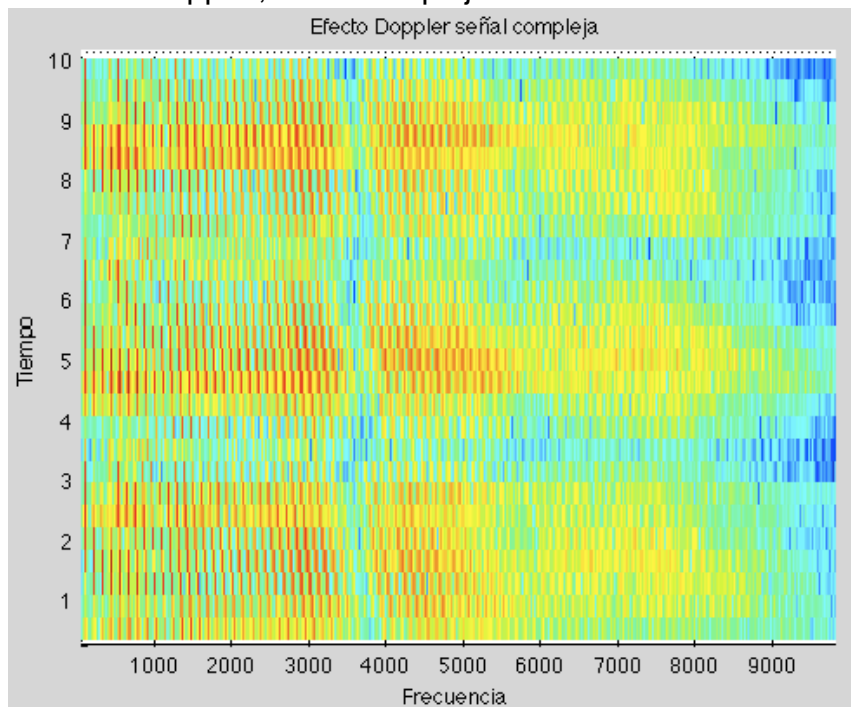
6.3.7 Medición del cambio Doppler obtenido con el *plug-in*, para una señal compleja, con movimiento sencillo en el eje Y. Se realizó la medición de una señal compleja para ver el cambio Doppler producido al utilizar el *plug-in*, se observa que el cambio es igual en todo el espectro ya que cambia la frecuencia fundamental y los armónicos de la señal en la misma proporción.

Figura 33. Vista de usuario del *plug-in* para la medición de movimiento en el eje Y. Señal compleja.



Interfaz de usuario.

Figura 34. Cambio Doppler, señal compleja

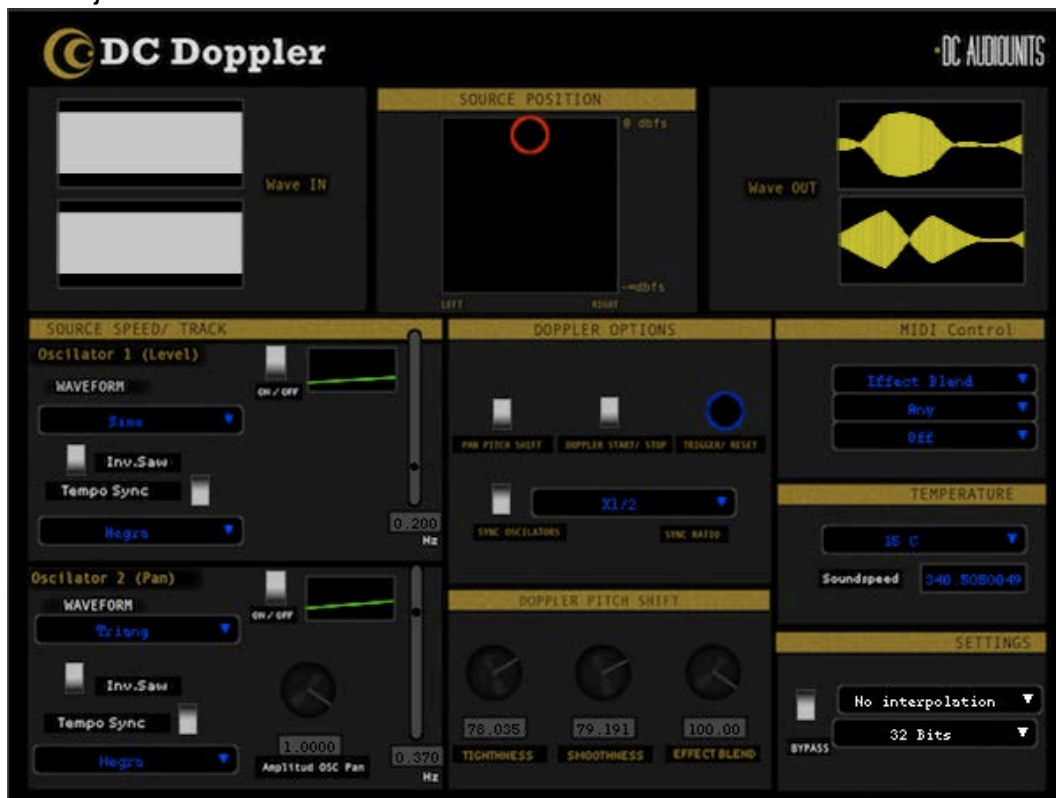


Herramienta de medición

La imagen 34 muestra el efecto Doppler para una señal compleja, se puede ver que el cambio es constante en todo el espectro, aunque no es posible determinar un intervalo exacto por la cantidad de información. El espectrograma está en escala lineal tanto en frecuencia como en tiempo, por esta razón se ve tanta información en alta frecuencia.

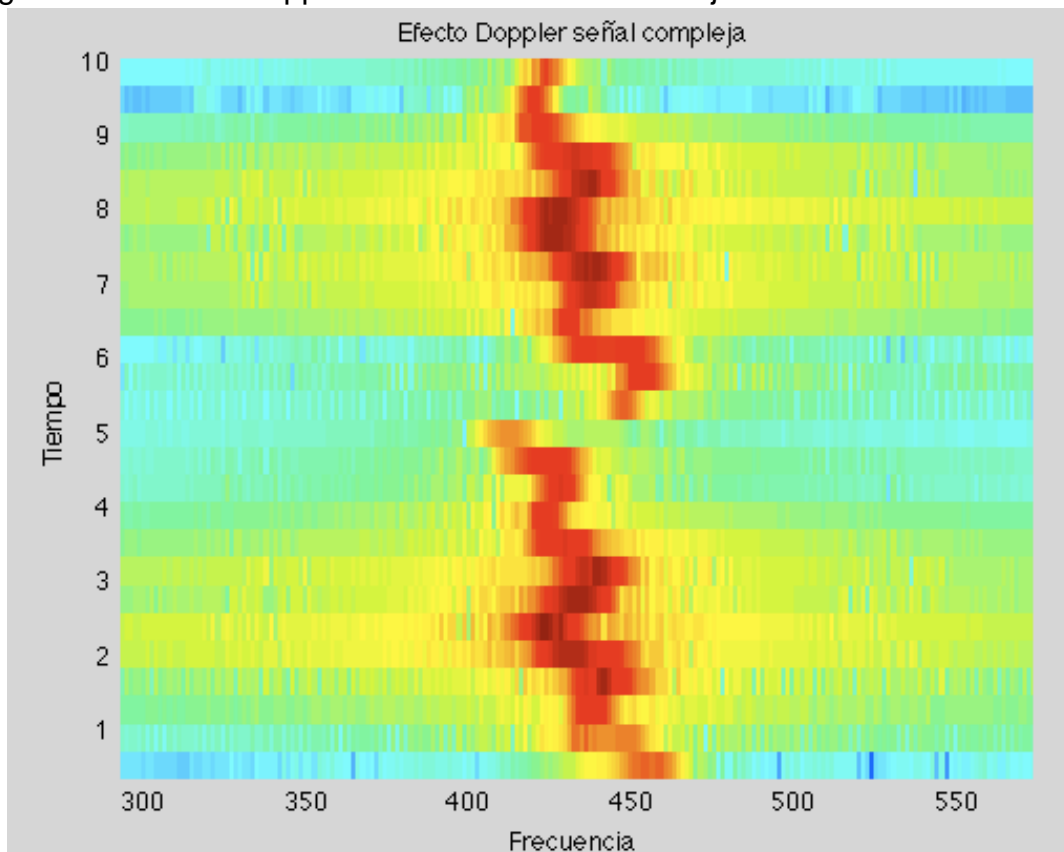
6.3.8 Medición del cambio Doppler para movimiento mixto en los dos ejes, X paneo, y Y nivel. Se midió el intervalo para un señal de 1000 Hertz, en un moviendo combinado en los dos ejes. La velocidad del oscilador fue de 0.2 Hertz senoidal, y a del oscilador 0.37 Hertz en una sola dirección. En este caso no fue posible determinar un intervalo exacto ya que este tipo de oscilación es aleatoria.

Figura 35. Vista de usuario del *plug-in* para la medición de movimiento mixto en los dos ejes.



Interfaz de usuario

Figura 36. Cambio Doppler movimiento mixto en 2 ejes



Herramienta de medición.

La grafica anterior muestra el cambio Doppler para una frecuencia de 440 Hertz por una trayectoria determinada por los dos osciladores. No se puede establecer un intervalo exacto ya que por no estar en sincronía los osciladores, la trayectoria es aleatoria.

6.4 Correlación de los intervalos Doppler a diferentes velocidades obtenidos con el *plug-in*, con los resultados obtenidos utilizando la ecuación.

Como resultado de la calibración que se le dio al *plug-in*, se midió el intervalo en Cents obtenido a diferentes velocidades en los osciladores de nivel y paneo. El intervalo obtenido a cada velocidad de oscilación es relacionado con la velocidad en kilómetros por hora en la que se produciría el mismo cambio acústicamente. Es necesario recordar que este análisis es válido solo para este *plug-in* y está basado en la calibración que se le dio.

Tabla 12. Relación en entre las velocidades del Oscilador 1, con velocidades Km/h, por medio del el cambio en cents.

Velocidad Osc 1	Intervalo	Km/h
0.01 Hz	2.22 Cents	1.57 km/h
0.02 Hz	4.44 Cents	3.14 km/h
0.03 Hz	6.66 Cents	4.71 km/h
0.04 Hz	8.88 Cents	6.27 km/h
0.05 Hz	11.1 Cents	7.84 km/h
0.06 Hz	13.27 Cents	9.36 km/h
0.07 Hz	15.54 Cents	10.95 km/h
0.08 Hz	17.76 Cents	12.51 km/h
0.09 Hz	19.99 Cents	14.07 km/h
0.1 Hz	22.21 Cents	15.62 km/h
0.2 Hz	44.42 Cents	31.05 km/h
0.3 Hz	66.63 Cents	46.28 km/h
0.4 Hz	88.87 Cents	61.34 km/h
0.5 Hz	111.06 Cents	76.17 km/h
0.6 Hz	133.27 Cents	90.17 km/h
0.7 Hz	155.48 Cents	105.29 km/h
0.8 Hz	177.69 Cents	119.57 km/h
0.9 Hz	199.9 Cents	133.67 km/h
1 Hz	222.1 Cents	147.59 km/h

Tabla 13. Relación en entre las velocidades del Oscilador 2, con velocidades Km/h, por medio del el cambio en cents.

Velocidad Osc 2	Intervalo	Km/h
0.01 Hz	2.22 Cents	1.57 km/h
0.02 Hz	4.44 Cents	3.14 km/h
0.03 Hz	6.66 Cents	4.71 km/h
0.04 Hz	8.88 Cents	6.27 km/h
0.05 Hz	11.1 Cents	7.84 km/h
0.06 Hz	13.27 Cents	9.36 km/h
0.07 Hz	15.54 Cents	10.95 km/h
0.08 Hz	17.76 Cents	12.51 km/h
0.09 Hz	19.99 Cents	14.07 km/h
0.1 Hz	22.21 Cents	15.62 km/h
0.2 Hz	44.42 Cents	31.05 km/h
0.3 Hz	66.63 Cents	46.28 km/h
0.4 Hz	88.87 Cents	61.34 km/h
0.5 Hz	111.06 Cents	76.17 km/h

7. CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

- Se concluyó que para emular el efecto Doppler de manera realista por medio de un *plug-in* sería necesario relacionar la velocidad de los osciladores, que también determinan la trayectoria de la fuente, utilizando un intervalo máximo de 222.12 para un desplazamiento en una sola dirección, en todo el espectro. Se calibro 0 dBfs como el nivel máximo que tendría una fuente a de 120 dB acústico a un metro del escucha, y por medio de la ley del inverso cuadrático se determinó el desplazamiento que emularía una oscilación

- Al analizar el desarrollo de *AudioUnits* en SonicBirth y Xcode, se concluye que la primera es una herramienta ideal para ingenieros de sonido o personas sin conocimientos avanzados en programación; Para desarrollar *audio units* o *plugins* por medio de código fuente en herramientas como XCode es necesario un dominio mucho más profundo de la programación enfocada a objetos. Se determinó SonicBirth como la herramienta para el desarrollo del *plug-in* de efecto Doppler en este proyecto.

- El *Audio Unit* emula el efecto Doppler, por medio de un movimiento oscilatorio de la señal, relacionando el nivel con la distancia, al irse acercando la fuente, o aumentando el nivel, el tono percibido ha aumentado y va disminuyendo a media que la fuente se acerca, al estar el nivel original de la señal, o en frente del escucha, el tono es el mismo al original, y al alejarse, o reducir el nivel, el tono sigue disminuyendo, desde el original. La adición de un movimiento sincronizado con el tempo del *host* le agrega un poco de musicalidad al efecto.

- La interfaz de usuario permite una manipulación adecuada de todos los parámetros del *plug-in*, se establecieron algunos *presets* y comprobaron las opciones de automatización y control MIDI, es necesario detener el motor de audio para la correcta manipulación de todos los parámetros.

- Se midió el cambio de tono producido por el *plug-in*, correlacionando los resultados con los datos obtenidos con los cálculos a diferentes velocidades de desplazamiento de la fuente, estableciendo una relación entre las velocidades de la fuente y del oscilador.

- El *audio unit* de efecto Doppler trabaja de manera adecuada tanto en AU Lab como en Garage Band sin sobrecargar la CPU en frecuencias de muestreo comúnmente utilizadas, y con un tiempo de procesamiento adecuado incluso

para su aplicación en tiempo real, aunque se necesita una CPU de gran capacidad para trabajar en frecuencias de muestreo altas.

7.2 RECOMENDACIONES

- Para desarrollo de *plug-ins* en XCode, con interfaz gráfica de usuario personalizada, se recomienda reunir un equipo de trabajo para distribuir las tareas necesarias para el desarrollo del *AudioUnit*, como podrían ser diseñador, desarrollador del código DSP y desarrollador del código para la GUI.

- Con la finalidad de determinar un proceso organizado en el desarrollo del algoritmo para el *audio unit*, es necesaria la previa realización de un boceto, un documento de función y una de tabla de parámetros, de esta manera obtiene más eficiencia y efectividad en el desarrollo del *plug-in*.

- Para emular el efecto Doppler por medio de un plugin dentro de un DAW es recomendado asumir que el receptor no se encuentra en movimiento.

- Se recomienda aplicar un cambio de tono en un intervalo máximo de 500 cents a la señal de audio para emular efecto el Doppler, y relacionar la velocidad de la fuente con el cambio de *pitch*. Se recomienda además establecer este cambio común cambio máximo, ascendente o descendente, ya que la formula no especifica el cambio en cada punto del movimiento de la fuente.

- Para facilitar el uso del plugin-in por parte de un usuario, es necesario realizar un "Guía de Uso", explicando el funcionamiento de sus parámetros, asignaciones MIDI y controles. Para este *audio unit* se incluye una en el anexo 5.

8. ERRORES SISTEMATICOS

- Se encontró un error en el funcionamiento del elemento PAD MIDI XY en SonicBirth, este no permite asignar diferentes rangos de valores para los ejes vertical y horizontal, y tampoco permite elegir diferentes unidades.
- Se pueden añadir ruidos al utilizar el *plug-in* al utilizar el *plug-in* a una resolución de 64 Bits, este presenta un funcionamiento mucho más estable a 32 bits.
- Es necesario detener el motor de trabajo para el correcto funcionamiento del audio unit, ya que al cambiar algunas de las opciones en los osciladores como la forma de onda o activar las velocidades sincronizadas, los osciladores salen de sincronía, o no es posible tener un funcionamiento óptimo en el *plug-in* de efecto Doppler.
- Se presenta inestabilidad a la hora de desarrollar audio units en SonicBirth, debido a que se cierra constantemente el programa de manera inesperada, es necesario guardar la sesión con frecuencia, y para poder deshacer alguna acción también es necesario grabar previamente.
- No es posible automatizar el parámetro "*keyboard tap*" de Sonic Bort, utilizado para controlar la opción *Trigger / Reset* del *audio unit* de efecto Doppler.

9. BIBLIOGRAFÍA

ALEXANDER ELLIS. On the Musical Scales of Various Nations. 1885.

ANDY DUNCAN, DAVE ROSSUM. Fundamentos para Pitch Shifting, Presentado en la 85 convención, 1988.

ANTONI CARRION ISBERT. Principios Básicos del Sonido, Diseño acústico de espacios arquitectónicos, 1998

APPLE INC. Audio Unit Programming Guide, 2007.

JOSEPH MACHLIS,. An Introduction to Perceptive Listening, The Enjoyment of Music. 1963.

LISA GOTHARD, JOE ROSEN. Encyclopedia of Physical Science. Doppler Effect. 2009. p. 155.

ROBERT RESNICK, DAVID HALLADAY,. Física, Ondas Sonoras. Efecto Doppler. 1974.

GLOSARIO

AMPLITUD En física la amplitud de un movimiento oscilatorio, ondulatorio o señal electromagnética es una medida de la variación máxima del desplazamiento u otra magnitud física que varía periódica o cuasi periódicamente en el tiempo. Es la distancia máxima entre el punto más alejado de una onda y el punto de equilibrio o medio.

API Interfaz de programación de aplicaciones (API) Una interfaz de programación de aplicaciones o API (del inglés *application programming interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

AUDIO UNIT Un Audio Unit (Unidad de Audio) comúnmente abrevia do como AU, es un “*plug-in*” para Mac OSX que agranda aplicaciones de audio digitales como Logic y Garage Band, también se pueden utilizar los Audio Units para desarrollar características propias para estas aplicaciones.

DAW Originado en el principio de los 80s, el término estación de trabajo de audio digital, se refiere a sistemas sin cinta análoga, basados en computadores como el New England Digital Synclavier y Fairlight, que usaba discos duros para almacenar datos.

EFECTO DE AUDIO Los efectos de audio son procesos, análogos y digitales que modifican las propiedades originales de una señal de audio, o agregan sonoridades a partir de su comportamiento original.

EFECTO DOPPLER: Cuando la fuente de ondas y el observador están en movimiento relativo con respecto al medio material en el cual la onda se propaga, la frecuencia de las ondas observadas es diferente de la frecuencia de las ondas emitidas por la fuente.

GENERADOR DE ONDA Un generador de señales, de funciones o de formas de onda es un dispositivo electrónico de laboratorio que genera patrones de señales periódicas o no periódicas tanto analógicas como digitales. Se emplea normalmente en el diseño, test y reparación de dispositivos electrónicos; aunque también puede tener usos artísticos.

INTERFAZ DE PROGRAMACIÓN DE APLICACIONES (API) Una interfaz de programación de aplicaciones o API (del inglés *application programming interface*) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción.

INTERFAZ GRÁFICA DE USUARIO Conocida también como **GUI** (del inglés *graphical user interface*) es un programa informático que actúa de interfaz de

usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.

PLUG-IN: Un *plug-in* es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

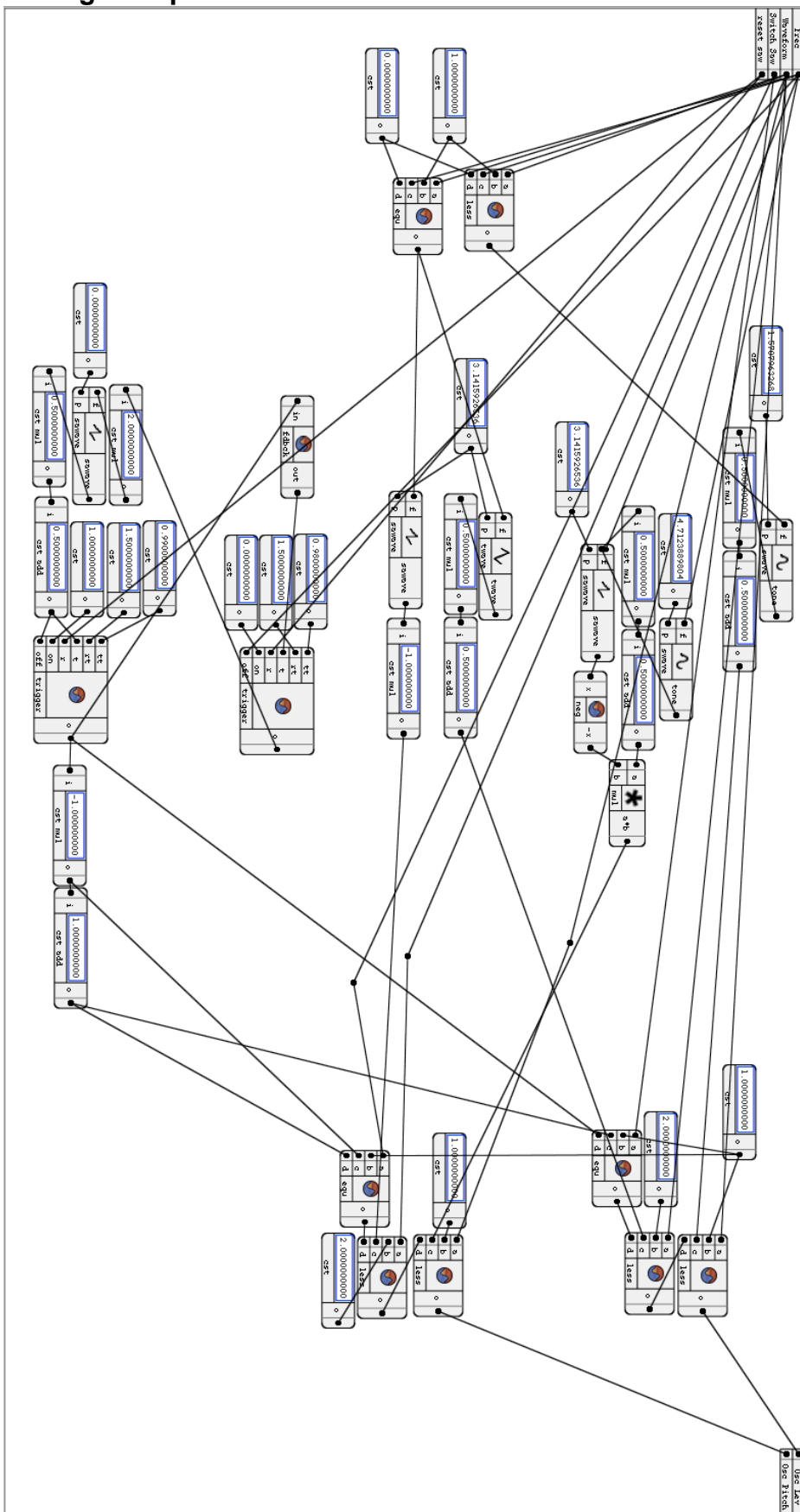
PROCESADOR DIGITAL DE SEÑALES O DSP Es un sistema basado en un procesador o microprocesador que posee un juego de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Debido a esto es especialmente útil para el procesamiento y representación de señales analógicas en tiempo real: en un sistema que trabaje de esta forma (tiempo real) se reciben muestras (samples en inglés), normalmente provenientes de un conversor analógico/digital (ADC).

PROGRAMACIÓN ORIENTADA A OBJETOS O POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.

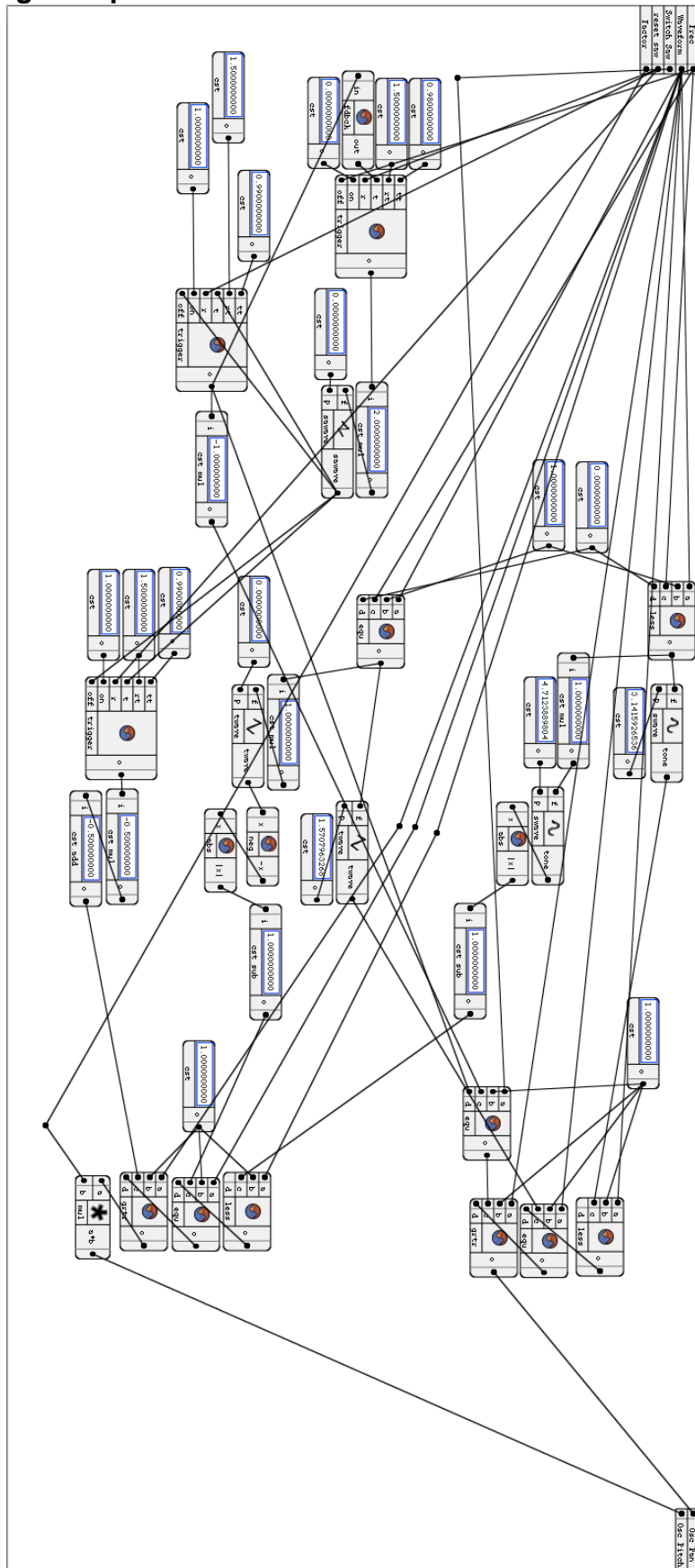
OSCILOSCOPIO Un osciloscopio es un instrumento de medición electrónico para la representación gráfica de señales eléctricas que pueden variar en el tiempo. Es muy usado en electrónica de señal, frecuentemente junto a un analizador de espectro.

TONO (PITCH) El pitch representa la frecuencia fundamental percibida en un sonido, es uno de los cuatro atributos principales de un sonido junto con nivel de presión sonora, timbre, y ubicación de la fuente sonora.

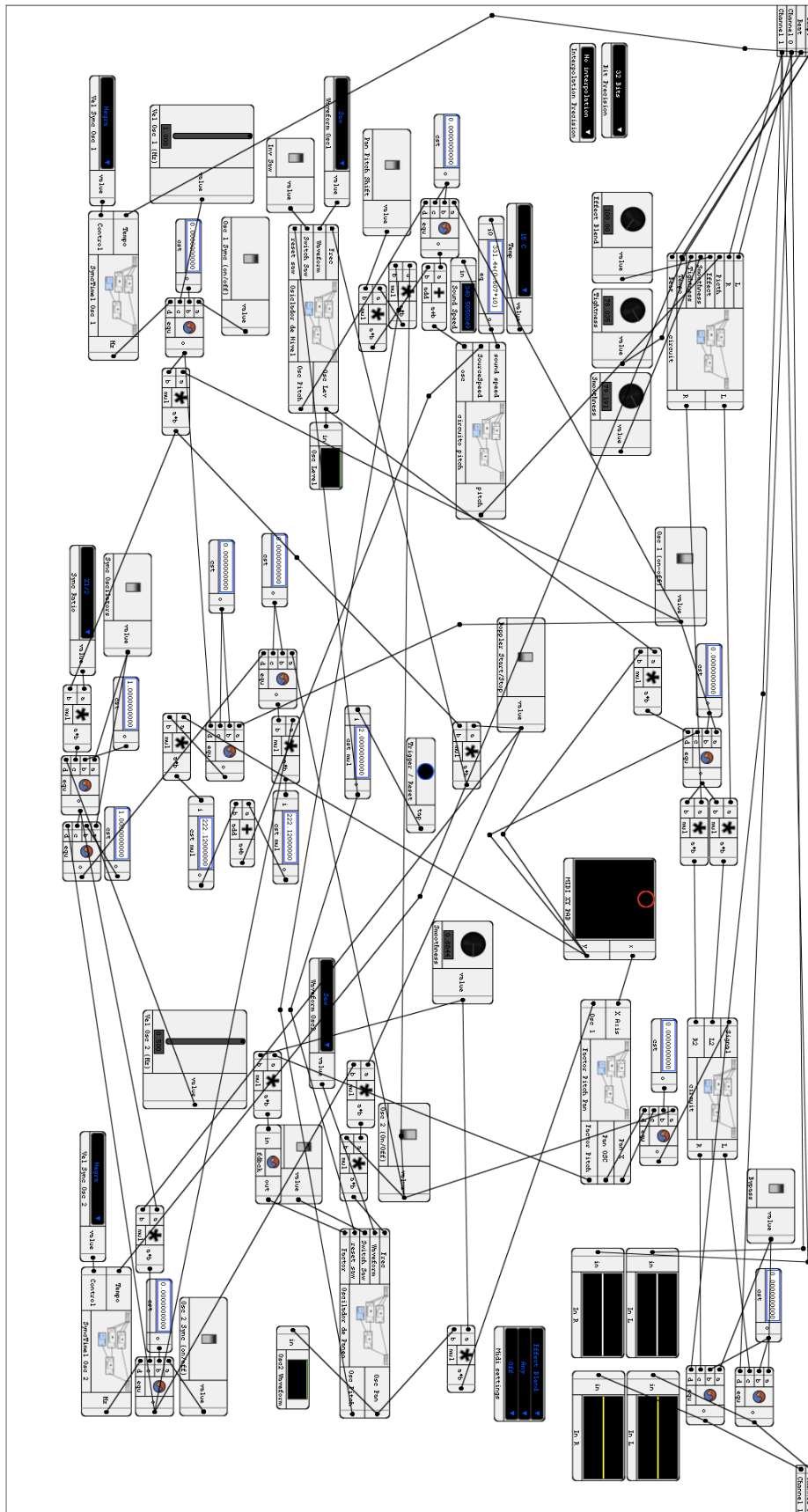
Anexo 1. Diagrama para Oscilador 1.



Anexo 2. Diagrama para Oscilador 2.



Anexo 3. Diagrama de bloques para el Audio Unit de Doppler Effect, en SonicBirth.



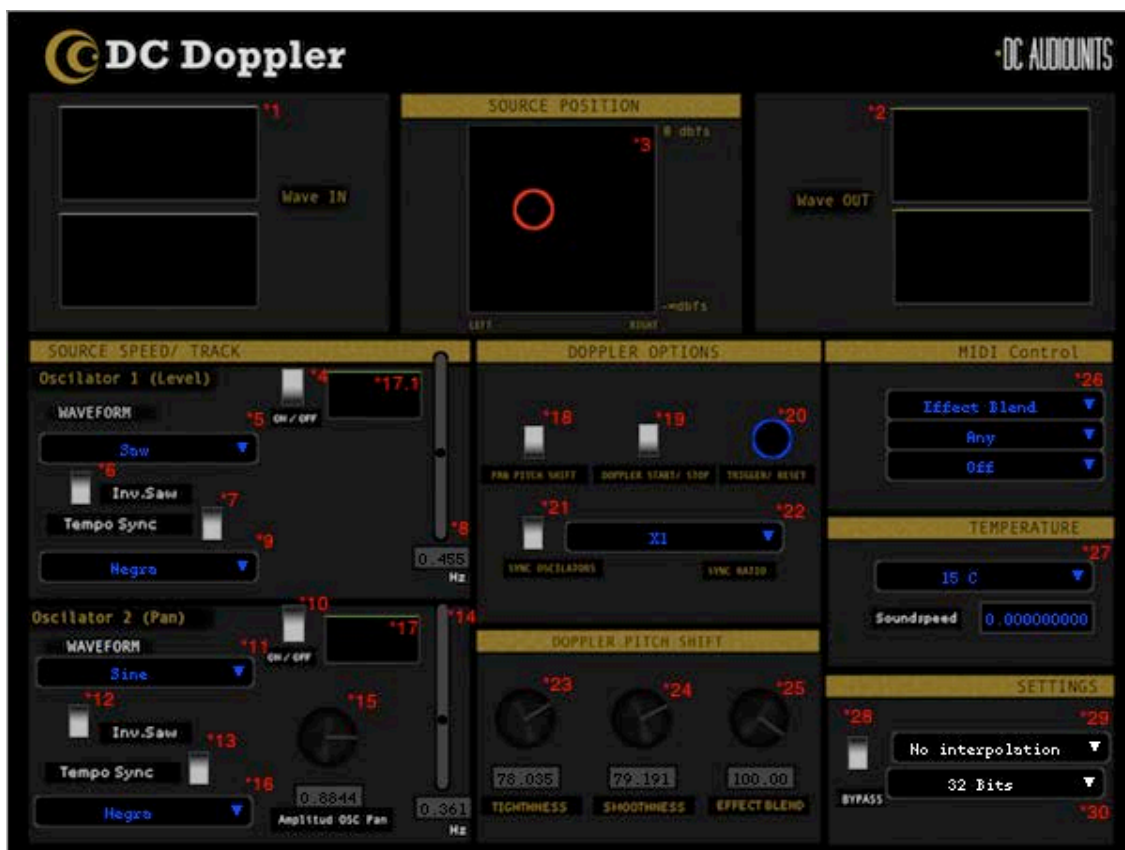
Anexo 4 Herramienta de medición en MatLab

```
clear all
clc

[x,fs]=wavread('Mix mono.wav');
L=length(x);
nfft=2048*16;%2^nextpow2(L);
x1=x(:,1);
win=round(L/4);
over=round(win/2);
%sound(x,fs)
spectrogram(x1,nfft,nfft/2,nfft,fs);
title('Efecto Doppler seÑal compleja')
ylabel('Tiempo')
xlabel('Frecuencia')
```

Anexo 5. Manual de Usuario

El *plug-in* DC Doppler de DC AudioUnits, realizara una emulación digital del “Efecto Doppler” a una señal deseada, basándose en las formulas descritas por el físico Christian Doppler, dando una sensación de movimiento dentro del espacio acústico por medio de un *plug-in*, que podrá ser utilizado en formato Audio Unit para Logic, Garage Band, y otros DAWs.



DESCRIPCIÓN DE ELEMENTOS Y PARÁMETROS

- *1 **Osciloscopio 1:** Muestra la señal de entrada
- *2 **Osciloscopio 2:** Muestra la señal de salida, procesada por el plug-in
- *3 **PAD MIDI XY:** Determina la posición de la fuente en los ejes X (Paneo) y Y (Nivel)

CONTROLES OSC 1 (EJE Y, O NIVEL)

***4 Switch OSC 1 (On/Off):** Activa o desactiva la oscilación de la señal en el eje Y (Nivel)

***5 Waveform 1:** Menú para elegir la forma de onda para el movimiento del OSC 1, se puede elegir entre senoidal triangular y Diente de Sierra.

***6 Inv Saw 1 (On/Off):** Switch para invertir la señal Saw del oscilador 1, en caso de ser seleccionada.

***7 Tempo Sync 1 (On/Off):** Switch para habilitar la sincronización de la velocidad de la oscilación en el eje Y, con el tempo de la sesión.

***8 Velocidad del oscilador 1 (Hz):** Determina la velocidad de la oscilación de la señal en el eje Y.

***9 Menú de Velocidades Sincronizadas 1:** Permite al usuario elegir diferentes velocidades sincronizadas para el oscilador 1, entre las opciones negra, blanca, redonda y tresillo.

CONTROLES OSC 2 (EJE X O PANEAO)

***10 Switch OSC 2 (On/Off):** Activa o desactiva la oscilación en el eje X (Posición Estéreo)

***11 Waveform 2:** Menú para elegir la forma de onda para el movimiento del OSC 2, se puede elegir entre senoidal triangular y Diente de Sierra.

***12 Inv Saw 2 (On/Off):** Switch para invertir la señal Saw del oscilador 2, en caso de ser seleccionada.

***13 Tempo Sync 2 (On/Off):** Switch para habilitar la sincronización de la velocidad de la oscilación en el eje X, con el tempo de la sesión.

***14 Velocidad del oscilador 2 (Hz):** Determina la velocidad de la oscilación de la fuente en el eje X.

***15 Amplitud OSC Paneo: (0-100%):** Determina la amplitud de la oscilación de la fuente en el eje X

***16 Menú de Velocidades Sincronizadas 2:** Permite al usuario elegir diferentes velocidades sincronizadas para el oscilador 2, entre las opciones negra, blanca, redonda y tresillo.

***17 Osciloscopio 3:** Muestra la manera en que se comporta el Oscilador 1

***17.1 Osciloscopio 4** Muestra la manera en que se comporta el Oscilador 2

DOPPLER OPTIONS

***18 Pan Pitch Shift (On/Off):** Activa o desactiva el cambio de pitch producido por el movimiento de la señal en el eje X(Posición Estéreo)

***19 Doppler Start Stop (On/Off):** Permite iniciar y detener el movimiento de los osciladores 1 y “

***20 Trigger/ Reset (Boton):** Controla el movimiento de la señal cuando la opción “Saw” está activada.

***21 Sync Oscilators (On/Off):** Sincroniza las velocidades de los osciladores 1 y 2 entre sí.

***22 Sync Ratio:** Determina la relación entre las velocidades de los osciladores 1 y 2 al estar sincronizados entre sí.

DOPPLER PITCH SHIFT

***23 Tightness: (0-100%):** Determina el ataque del cambio de pitch al ser procesada la señal.

***24 Smoothness (0-100%):** Suaviza el cambio de pitch producido por el desplazamiento de la fuente.

***25 Efect Blend (0-100%):** Determina el grado de diferenciación entra el

SETTINGS

***26 MIDI Settings:** Menú para configurar el Control MIDI para el AudioUnit, se determina para cada control el canal MIDI, y el parámetro MIDI se utilizara

***27 Temperature:** Determina la velocidad de simulación estableciendo el factor en que esta afecta el cambio de pitch

***28:Bypass (On/Off):** Permite comprar la señal original con la procesada.

***29 Interpolation:** Determina si los elementos trabajan de manera interpolada o no.

***30 Bit Resloution:** Determina la resolución de procesamiento en bits, 32 o 64.

Anexo 6. Circuito del Reverberador de Schoeder, desarrollado en "SonicBirth"

